UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

CSVMS - Cyber Security Vulnerability Management System

João Rafael Xisto Miranda

Mestrado em Segurança Informática

Trabalho de projeto orientado por:
Prof. Doutor Nuno Fuentecilla Maia Ferreira Neves

2020

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

CSVMS - Cyber Security Vulnerability Management System

João Rafael Xisto Miranda

Mestrado em Segurança Informática

Trabalho de projeto orientado por Prof. Doutor Nuno Fuentecilla Maia Ferreira Neves e supervisionado na Altice por Eng. José António dos Santos Alegria.

2020

Agradecimentos

Primeiramente, gostava de agradecer aos meus pais, Josué e Paula, por tudo o que fizeram por mim. Todos os valores, educação e oportunidades que me proporcionaram, e que me definem hoje como pessoa. Sem a vossa persistência e vontade de ser mais, nada disto seria possível. O maior agradecimento é sem dúvida para vocês e nunca poderei descrever por palavras o quão grato me sinto.

Gostava de agradecer também a toda a equipa da DCY, especialmente à Ana Paula Gonçalves que foi o meu suporte principal durante o projeto. Sem a tua ajuda nada disto seria possível, foste incansável e essencial tanto na minha integração na Altice como em todas as fases do projeto.

Agradeço ao meu orientador, Professor Nuno Neves, por toda a sua disponibilidade, preocupação, paciência e conhecimento transmitido durante este processo. Sinto-me muito grato por ter a oportunidade de ser orientado pela minha principal referência no mundo académico da informática. É sem dúvida uma inspiração para querer aprender mais.

Agradeço também ao meu co-orientador, Engº José Alegria. Foi uma das pessoas mais importantes neste percurso, não só por toda a coordenação do projeto, mas também por todos os valores que transmite dia após dia. Agradeço-lhe por todas as intervenções e palavras que me deu ao longo do ano, todas as conversas que recordo e que me inspiram para ser melhor pessoa e melhor profissional. É certamente uma das pessoas mais rigorosas e justas com que tive o prazer de me cruzar, e fico muito grato pelo privilégio de ser seu mestrando.

Agradeço à Sara Nascimento por tudo, literalmente por tudo. Não esqueço o que fizeste por mim antes, durante e depois de concretizar o projeto. Foste incansável e sobretudo amiga, muito obrigado. Não esquecendo o Gonçalo, o Fábio e Mariana, que me acolheram e que estiveram sempre disponíveis para me ajudar. Muito obrigado à Beatriz, à Cátia, à Inês e ao Zé por fazerem este percurso comigo. Tive o privilégio de ter-vos como colegas e amigos para partilhar esta experiência, e levo memórias para sempre.

Gostava também de agradecer a todos os meus amigos. Tanto aos que conheci na FCUL e que me acompanharam durante este grande desafio académico, como aos que apareceram pelas mais diversas circunstâncias. Acreditem em vocês, sejam mais e melhor!

Last but not least, I want to thank one of my biggest inspirations for my whole development as a person. Peter, I can't describe how grateful I am for everything you keep teaching me. Dream big you said, so here we go!

A todos os que me inspiraram a acreditar.

Resumo

A monitorização contínua das vulnerabilidades presentes nos sistemas que compõem a infraestrutura de uma organização, é um processo fundamental para garantir segurança e estabilidade dos seus serviços. A complexidade deste processo é proporcional à diversidade da infraestrutura, sendo por vezes um fator que pode causar dificuldades em empresas de grandes dimensões, especialmente na priorização do processo de remediação. Com a utilização de diferentes tecnologias é importante considerar a necessidade de diversos métodos de análise, que produzem informação distinta. Esta informação deve ser armazenada de forma organizada,*i.e*, de forma canónica e unificando as diferentes taxonomias usadas pelas diferentes ferramentas de descoberta de vulnerabilidades. Deste modo é possível garantir uma análise eficiente da informação e consequentemente facilitar o processo de resolução ou remediação dos problemas encontrados.

Com o objetivo de agilizar esta tarefa, reduzindo drasticamente o tempo de análise e resolução de vulnerabilidades, foi desenvolvido o *Cyber Security Vulnerability Management System* (CSVMS). O CSVMS é um sistema que coordena automaticamente a extração, processamento e armazenamento da informação sobre as vulnerabilidades existentes. Esta informação é enriquecida e guardada num repositório, seguindo uma estrutura normalizada e transversal à sua origem. Deste repositório é possível extrair relatórios operacionais que serão entregues às entidades competentes, beneficiando diretamente as equipas responsáveis pelos ativos com informação útil e priorizada.

A informação produzida pelo CSVMS pode ser consultada de acordo com o objetivo pretendido. Podem ser analisados os relatórios operacionais, gerados automaticamente e que permitem realizar consultas dinâmicas sobre os ativos em geral. Estes são interativos e adaptam a informação de acordo com o foco de investigação do analista, através da aplicação de filtros. É também dada a possibilidade de consulta da informação armazenada em bruto. Este método de consulta permite analisar em detalhe cada vulnerabilidade específica, contendo toda a informação recolhida pelas múltiplas ferramentas usadas. Deste modo, limita-se o processo de investigação num só local, que armazena toda a informação de forma segura. Permitindo poupar tempo em tarefas desnecessárias e não existir dependência da consulta a terceiros.

O CSVMS foi implementado sobre a infraestrutura operacional da MEO, processando a informação recolhida sobre este ambiente. Por fim foi avaliada a eficácia do sistema, através da realização de inquéritos e da adição de uma nova ferramenta, produzindo resultados muito satisfatórios.

Palavras-chave: monitorização, vulnerabilidades, gestão, ativos, consulta

Abstract

In the current global environment, the risks associated with cybersecurity threats are one of the most important concerns for any organization. Those concerns tend to increase due to the continuous evolution of industries, motivating the rush for new technological approaches that may bring more business prosperity and expansion. This evergoing process is responsible for the growth of new opportunities, which can highly benefit the company but may attract malicious actors that want to exploit any vulnerability left behind. So, for every vulnerability left in the system, the risk of being attacked increases dramatically, resulting in substantial financial losses, reputational damage and information leaks that might be enough to sentence the end of a company.

The management of the infrastructure that supports the business is crucial for companies to assure the quality and reliability of its services, as well as the security of the information held. This process covers the analysis of all the required elements for the company to run and might be very complex.

First, companies have to scan every possible asset to make sure that most of the existing vulnerabilities are identified. It commonly requires the usage of different tools for specific scanning and analysis to get the best results on different groups of assets. With the results obtained from the scanning phase, a list of existing vulnerabilities is created, which the cybersecurity technicians have to analyze to make sure that most of the vulnerabilities are fixed as soon as possible. These steps end up being very time-consuming due to the heterogeneity of the tools and also by the possibility of error occurance during the manual analysis of all the data.

According to the current vulnerability assessment procedure, adopted by MEO, the results produced by all the scanning tools are gathered by the cybersecurity analysts to be inspected. After the detailed study of all the vulnerable environment, vulnerabilities start to get fixed based on its priority (risk/cost ratio). The inefficiency of this process will have an extreme impact on such a time-critical task, preventing the optimization of the company's resilience.

Since this procedure is not as efficient as desired, we created the Cyber Security Vulnerability Management System (CSVMS), an effective time-saving vulnerability management system. This solution automatically extracts, processes, and delivers the information gathered from every scanning tool used internally. The aim is to reduce the time spent on manual extraction and organization of the required vulnerability assessment information, allowing the technicians to save time on unnecessary and error-prone tasks. The CSVMS aggregates and normalizes the collected information, creating a consolidated report to analyze the data in a time-sensitive manner. This generated report presents the simplified and enriched version of the relevant data, allowing for the dynamic inspection and resolution of the issues found.

After including CSVMS in MEO's operational routine, an evaluation was undertaken through feedback from the cybersecurity analysts team. In general, the results were very positive, proving that the

system is very adaptable and provides helpful normalized information through a user-friendly environment. These evaluation conclusions are relevant to assure the best quality of the system results, ending up by saving essential time that, in the end, can be used to turn the company even more secure.

Keywords: evolution, vulnerability, management, optimization, assets

Conteúdo

Lista de Figuras **xviii**

Lista de Tabelas **xxi**

1 Introdução **1**

1.1 Motivação . . . 2
1.2 Objectivos . . . 3
1.3 Contribuições . . . 4
1.4 Estrutura do documento . . . 5

2 Contexto e Trabalho Relacionado **7**

2.1 Princípios de um ataque . . . 7
2.2 Vulnerabilidade . . . 8
2.2.1 Potencial impacto de uma vulnerabilidade . . . 8
2.2.2 Ciclo de vida das vulnerabilidades . . . 9
2.2.3 CVE . . . 11
2.2.4 CVSS . . . 11
2.3 Motores de deteção e análise de vulnerabilidades . . . 13
2.3.1 Superfícies de análise . . . 15
2.3.2 Área de incidência . . . 15
2.3.3 Processo de funcionamento . . . 16
2.3.4 Correlação da informação de várias ferramentas . . . 17
2.4 Ferramentas utilizadas . . . 18
2.4.1 Cycognito . . . 18
2.4.2 Qualys . . . 19
2.4.3 Bitsight . . . 20
2.4.4 Mozilla Observatory . . . 21
2.4.5 Processo de gestão de vulnerabilidades . . . 22
2.4.6 Projetos relacionados . . . 24
2.5 Conclusão . . . 25

3 Estrutura do CSVMS **27**

3.1 Requisitos do projeto . . . 27
3.2 MDAV'S incluídos no projeto . . . 28

3.3 Modelo do projeto . . . 29
3.3.1 Motor de calendarização . . . 30
3.3.2 Motor de processamento . . . 31
3.3.3 Motor de visualização . . . 38
3.4 Conclusão . . . 39

4 Implementação **41**
4.1 Motor de calendarização . . . 41
4.2 Motor de processamento . . . 42
4.2.1 Relatório de agregação . . . 43
4.2.2 Sub-módulo - Implementação dos processos de extração do *Qualys* e *Bitsight* . . . 44
4.2.3 Sub-módulo - Implementação do processo de extração *Cycognito* . . . 45
4.2.4 Sub-módulo - Componente de processamento . . . 47
4.2.5 Cálculo do risco dos ativos . . . 49
4.2.6 Implementação do armazenamento de informação na base de dados . . . 49
4.2.7 Implementação do *software* . . . 50
4.3 Motor de visualização . . . 51
4.4 Conclusão . . . 53

5 Resultados e Avaliação **55**
5.1 Questionário de usabilidade . . . 55
5.1.1 Participantes . . . 55
5.1.2 Resultados . . . 56
5.1.3 Discussão . . . 58
5.2 Implementação do novo caso de uso . . . 58
5.2.1 Implementação . . . 59
5.2.2 Tempo decorrido . . . 59
5.2.3 Discussão . . . 60
5.3 Conclusão . . . 61

6 Conclusão e Trabalho Futuro **63**
6.1 Conclusão . . . 63
6.2 Trabalho futuro . . . 63

A Relatório operacional **65**
A.1 Exemplo de relatório operacional - parte 1 . . . 66
A.2 Exemplo de relatório operacional - parte 2 . . . 67

B Questionário de usabilidade **69**
B.1 Questionário de usabilidade adotado do modelo SUS . . . 69

Abreviaturas **71**

Bibliografia **72**

Lista de Figuras

1.1 Representação do atual processo de gestão de vulnerabilidades 3
1.2 Resultado esperado com a concretização do projeto. 4

2.1 Modelo AVI 8
2.2 Ciclo de vida das vulnerabilidades 10
2.3 CVSS - Grupo de métricas base 12
2.4 CVSS - Grupos de métricas temporais e de ambiente 12
2.5 Arquitetura de um MDAV 14
2.6 Processo de funcionamento de um MDAV 16
2.7 Excerto de relatório executivo produzido pelo Cycognito 17
2.8 Exemplo de relatório técnico extraído da plataforma Cycognito. 19
2.9 Exemplo de relatório técnico extraído da plataforma Qualys. 20
2.10 Excerto de relatório extraído da plataforma Bitsight. 21
2.11 Excerto de relatório extraído da plataforma *Mozilla Observatory*. 21
2.12 Ciclo de funcionamento do processo de gestão de vulnerabilidades 22
2.13 Representação da janela temporal em que é desconhecida a vulnerabilidade dos ativos a uma ameaça já pública. 23

3.1 Arquitetura do CSVMS 30
3.2 Motor de calendarização 31
3.3 Arquitetura do motor de processamento 31
3.4 Relatório de agregação de informação 32
3.5 Relação entre as categorias de informação e o seu âmbito informativo 33
3.6 Estrutura do sub-módulo 34
3.7 Caso de uso genérico do processo de extração de informação 35
3.8 Centralização da informação previamente dispersa num só documento. 36
3.9 Representação do motor de visualização 39

4.1 Representação do motor de calendarização 42
4.2 Etapas do processo realizado pelo motor de processamento 42
4.3 Exemplo de funcionamento do relatório de agregação 44
4.4 Casos de uso da extração de informação do Qualys e *Bitsight* 44
4.5 Caso de uso da extração de informação do Cycognito 46
4.6 Exemplo do processamento da informação de um MDAV 47

4.7 Modelo de classificação normalizado. 48
4.8 Diagrama de classes do motor de processamento 50
4.9 Resultados presentes no índice do projeto CSVMS 51
4.10 *Visualizations* criadas para representar dados 52
4.11 Representação integral da informação de uma única vulnerabilidade no *Kibana* 52

5.1 Distribuição da pontuação SUS por participante. 57
5.2 Distribuição da pontuação SUS por questão. 57

Lista de Tabelas

2.1 Associação das classificações com a pontuação CVSS 13

3.1 Motores de deteção e analise de vulnerabilidades implementados no projeto. 29
3.2 Motores de deteção e analise de vulnerabilidades utilizados internamente mas que não foram implementados no projeto. 29
3.3 Motores de deteção e analise de vulnerabilidades que não chegaram a ser implementados internamente à MEO, não podendo ser usados no projeto. 29
3.4 Representação das categorias de risco de acordo com o risco bruto calculado. 38

5.1 *Heatmap* de distribuição de respostas do Questionário SUS. 56
5.2 Registo do tempo decorrido durante a implementação do novo caso de uso. 60

Capítulo 1

Introdução

Hoje em dia, grandes empresas dependem diretamente da informatização de processos para assegurar a estabilidade do seu funcionamento. São geridas e armazenadas quantidades significativas de dados altamente valiosos, que torna essencial a adoção das melhores práticas com o objetivo de garantir a sua segurança.

Um dos procedimentos mais importantes para garantir a segurança da informação é a monitorização contínua dos ativos. Só através da realização desta tarefa é possível conhecer realmente o estado dos sistemas e as suas fragilidades. Posteriormente podem ser tomadas medidas para mitigar as vulnerabilidades encontradas e tornar a infraestrutura mais robusta. No entanto, as infraestruturas das organizações são altamente heterogéneas. Tipicamente constituídas por diversos sistemas, com vários ambientes de trabalho que cooperam entre si. Esta premissa é relevante no processo de gestão dos ativos, dado que, não só torna mais complexa a análise da infraestrutura como também aumenta a superfície de ataque para agentes maliciosos.

Diferentes sistemas têm habitualmente vulnerabilidades distintas associadas. Estas vulnerabilidades derivam sobretudo das características de implementação e estrutura do sistema. Ou seja, quanto maior for a diversidade tecnológica, maior será a probabilidade de existirem vulnerabilidades. Para auxiliar a monitorização dos ativos, tipicamente são utilizadas ferramentas de análise automática de vulnerabilidades, pois a análise manual é um processo demorado, exaustivo e suscetível a falhas.

Cada ferramenta tem um domínio de análise específico, e executa varrimentos de testes sobre os ativos que se pretendem analisar. Embora os domínios sejam vastos, é comum a análise de vulnerabilidades em aplicações Web, em servidores que se encontram expostos ao exterior e servidores que mantêm operacionais serviços críticos. Para garantir abrangência de análise em pontos importantes, é feita uma gestão da capacidade de análise que prioriza os pontos de acesso do exterior. O objetivo é garantir que estes pontos de acesso são tão seguros quanto possível, reduzindo a probabilidade de ocorrerem intrusões.

Através da análise das respostas produzidas pelos varrimentos, é deduzido se um determinado ativo se encontra vulnerável ou não. Após esta etapa, são armazenados os resultados que serão posteriormente enriquecidos com informação acerca das vulnerabilidades encontradas (presente em repositórios de conhecimento das próprias ferramentas). Para finalizar, são produzidos relatórios que especificam os problemas encontrados, bem como possíveis soluções e informação sobre as vulnerabilidades. Esta abordagem pode ser extremamente ineficiente devido aos recursos que é necessário despender para analisar cada relatório.

O objetivo deste projeto é a criação de um sistema que promove a automatização da gestão de vulnerabilidades nos ativos da Altice Portugal (MEO), o Cyber Security Vulnerability Management System (CSVMS). Este feito é atingido com o aperfeiçoamento de cada uma das etapas do processo, desde a recolha de informação até à apresentação de resultados. Deste modo será possível corrigir todos os fatores que tornam ineficiente o atual procedimento. Como ponto de partida, abordaram-se as tarefas de recolha e armazenamento local da informação. Assim, simplifica-se a monitorização da infraestrutura, usufruindo da organização da informação.

Este projeto foi realizado na MEO, tirando partido do atual procedimento de gestão de vulnerabilidades, bem como das ferramentas utilizadas na empresa. Sendo a MEO uma das principais provedoras de serviços de telecomunicações em Portugal, é essencial definir um sistema eficiente que facilite a monitorização dos inúmeros de ativos.

1.1 Motivação

A frequência de ataques informáticos tem aumentado significativamente nos últimos anos. Este aumento é justificado com o acréscimo da adoção tecnológica para satisfazer os modelos de negócio e também com os potenciais ganhos com o roubo de informação. Segundo a *Gartner* [18], estima-se que em 2019 o investimento global na deteção, resposta e cuidados de privacidade rondou os 124 mil milhões de dólares. Este valor corresponde a um aumento de aproximadamente 8,7% em relação ao ano anterior, o que fornece uma indicação da relevância desta área na sociedade.

Segundo o atual processo de gestão de vulnerabilidades na MEO, o departamento de segurança é responsável por programar e executar os varrimentos, analisando posteriormente os resultados obtidos. Este processo está ilustrado na Figura 1.1, verificando-se a necessidade de inspeção de diferentes relatórios produzidos por várias ferramentas. Após esta análise, é feita uma agregação de vulnerabilidades por departamento e são preparados relatórios distintos, enriquecidos com informação de outras fontes. Cada relatório descreve as fragilidades encontradas nos ativos de um determinado departamento. Por fim, são enviados os relatórios para os respetivos destinatários, para que estes procedam à correção das fragilidades, se assim for possível.

Este processo é complexo, demorado e altamente suscetível a erros. É frequente a ocorrência de falsos positivos, situações onde são reportadas vulnerabilidades que não existem, degradando ainda mais o seu desempenho. Um cenário comum é a deteção de uma vulnerabilidade, que é reportada e corrigida com um *workaround*. No entanto, ao analisar com outra ferramenta, a vulnerabilidade é novamente identificada pois a solução empregue não foi a mais adequada. Estes aspetos têm impacto direto no funcionamento dos departamentos, desperdiçando recursos e tempo que são necessários para realizar outras atividades.

Sendo esta gestão maioritariamente realizada sobre ativos críticos da MEO, o processo de análise atual não é de todo o processo mais eficaz. O seu desempenho condiciona o fator de ciber higiene da empresa, que seria significativamente maior com uma monitorização mais rápida e abrangente. A ciber higiene representa os cuidados básicos de caráter preventivo de segurança no ciberespaço. Portanto, se forem analisados mais ativos num espaço de tempo mais curto, será possível contribuir para a resolução mais rápida de vulnerabilidades. De acordo com o artigo da *Cloud Security Alliance* [13], a cultura de

ciber segurança nas organizações é um fator cada vez mais importante, tendo que ser proativa dado o aumento de circunstâncias que motivam à prática de ataques altamente devastadores. Assim sendo, a motivação para a realização deste projeto incide no problema acima descrito. Com a automatização do processo atualmente praticado, será possível analisar mais ativos e realizar mais tarefas utilizando os mesmos recursos. Estes benefícios tiram partido do tempo ganho nas várias etapas do processo. No fim, é esperado que a empresa seja beneficiada num todo, aumentando o seu índice de ciber higiene.

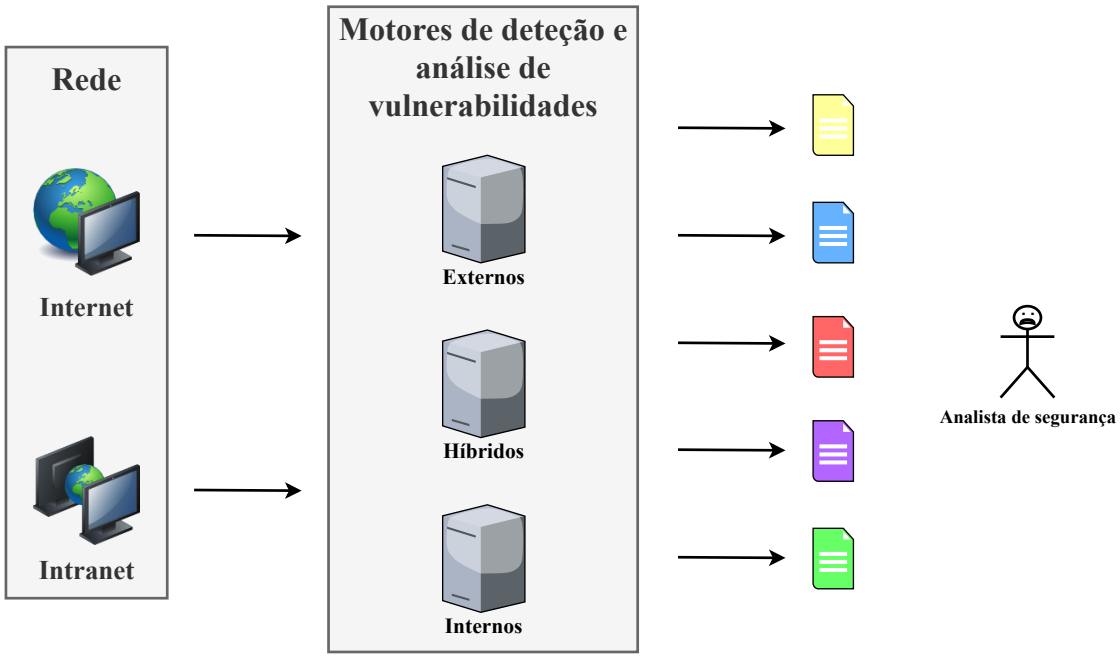


Figura 1.1: Representação do atual processo de gestão de vulnerabilidades

1.2 Objectivos

Da vasta gama de vulnerabilidades presentes na infraestrutura de uma grande organização, nem todas podem ser consideradas com a mesma prioridade. Existem vários fatores que influenciam a importância de cada vulnerabilidade, incluindo o seu índice de criticidade, a facilidade com que é possível de mitigar, as suas dependências e também quais os dispositivos afetados. Considerando que é boa prática verificar periodicamente os sistemas para identificar fragilidades, depois de identificadas é também necessário saber o que solucionar primeiro. A resolução torna-se mais fácil se houver disponível uma maior quantidade de informação acerca do problema.

Como objetivos deste projeto, primeiro, pretendemos compreender o domínio de cada ferramenta de varrimento, bem como que ativos são analisados. Assim, torna-se possível definir um esquema que permita mapear a forma como estas ferramentas se podem complementar e também, qual o melhor partido que se pode tirar da sua utilização. Este esquema define categoricamente a informação de cada ferramenta, criando relações entre os diversos parâmetros. Deste modo, torna-se possível a correlação de informação sobre as mesmas vulnerabilidades detetadas em ferramentas diferentes.

Após ser bem definida a relação entre as ferramentas, queremos perceber qual é a calendarização mais eficaz para a extração de relatórios. Através deste procedimento, esperamos conseguir definir as melhores rotinas de recolha de relatórios individuais, que causem o menor impacto possível sobre o serviço. Esta

é uma tarefa não trivial dada a heterogeneidade de modos de funcionamento e dos ativos a analisar. Alguns destes suportam serviços que podem sofrer problemas de disponibilidade com varrimentos mal calendarizados.

Com estas bases pretende-se criar um sistema que automatize a extração e priorização da informação. Esta informação após ser normalizada é armazenada num local centralizado, evitando a dispersão de dados. Para além disto, poderão ser extraídos relatórios operacionais, baseados na informação recolhida de acordo com os ativos. Atinge-se assim o objetivo final de dinamizar o processo de gestão de vulnerabilidades sobre os ativos da MEO (Figura 1.2).

Em suma, com este projeto pretende-se:

- Automatizar o processo de extração de relatórios;
- Reduzir a dispersão que existe atualmente da informação recolhida;
- Facilitar a análise e agregação da informação por departamento;
- Tornar mais prática a produção de relatórios operacionais úteis.

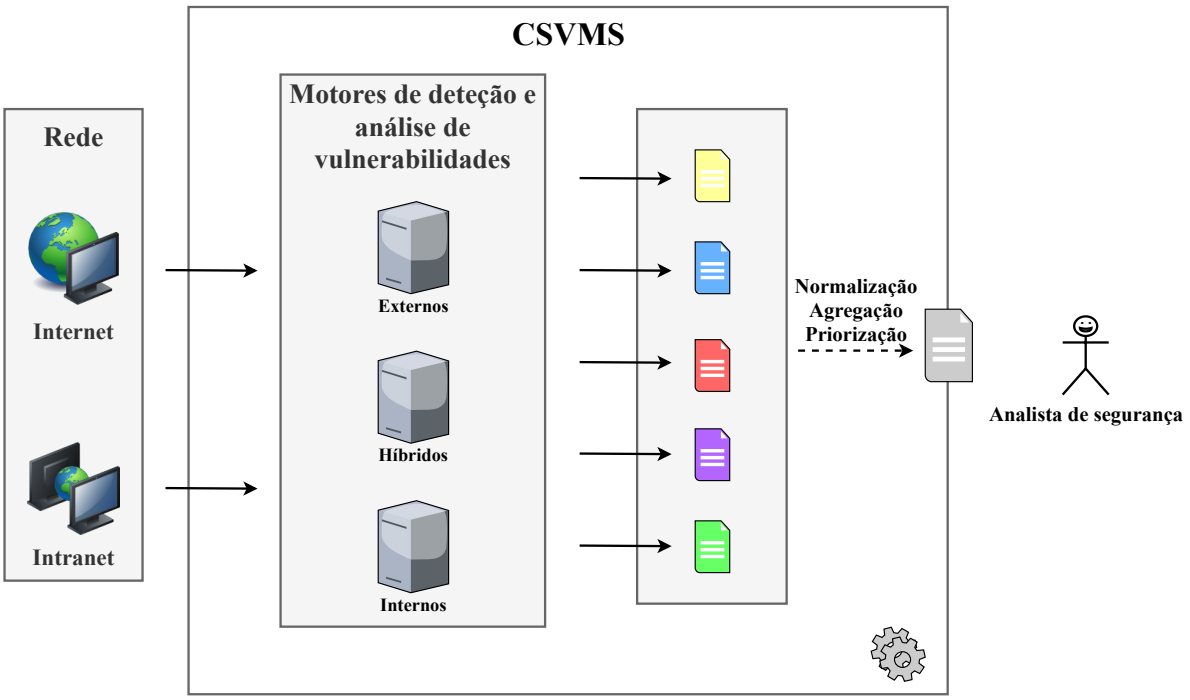


Figura 1.2: Resultado esperado com a concretização do projeto.

1.3 Contribuições

A grande contribuição deste projeto, consiste na investigação e desenvolvimento de um método mais eficiente para o processo de gestão de vulnerabilidades na MEO.

Este método foi concretizado num sistema que é responsável por auxiliar no processo de gestão de vulnerabilidades, e foca-se especialmente nos seguintes pontos:

- **Automatizar** a extração da informação sobre vulnerabilidades e assim minimizar o tempo despendido nesta tarefa;

- **Processar** a informação recolhida para que seja possível agregar informação independentemente da sua origem;
- **Enriquecer** a informação processada de forma a tornar o repositório do CSVMS tão completo quanto possível;
- **Disponibilizar** a informação de forma clara e prática para que possa ser facilmente consultada por técnicos.

1.4 Estrutura do documento

O resto deste documento está organizada da seguinte forma:

- Capítulo 2 : Contexto e Trabalho Relacionado - É apresentado o estado atual da arte nas áreas relacionadas com o projeto. Primeiro são apresentados todos os conceitos base, necessários para compreender os objetivos do CSVMS, e por fim são descritos outros projetos já existentes e os respetivos resultados obtidos.
- Capítulo 3 : Estrutura do CSVMS - É apresentada a arquitetura do sistema, fundamentando as decisões que foram tomadas durante o processo de planeamento do CSVMS;
- Capítulo 4 : Implementação - Neste capítulo são descritos em detalhe os métodos utilizados para implementar o sistema;
- Capítulo 5 : Resultados e Avaliação - É apresentada uma discussão acerca da avaliação realizada ao CSVMS. Esta avaliação envolve a experiência de utilização e a interação técnica com o sistema.
- Capítulo 6 : Conclusão - Este capítulo sumariza todo trabalho realizado na criação do CSVMS e apresenta conclusões finais sobre os resultados obtidos. São também introduzidas outras direções interessantes para investigação futura.

Capítulo 2

Contexto e Trabalho Relacionado

O projeto desenvolvido visa automatizar e dinamizar a gestão de vulnerabilidades em ambientes com elevada heterogeneidade de informação. A gestão de vulnerabilidades envolve a análise e classificação das vulnerabilidades detetadas através da monitorização contínua dos ativos. Neste capítulo é feita a descrição dos diversos conceitos teóricos e práticos que são fundamentais para a contextualização ao tema. Para cada conceito é explicado o processo comum de funcionamento e também outros subconceitos que são relevantes. Por fim, é feita a exposição das ferramentas que são utilizadas no projeto. Estas ferramentas são a base para a análise automática dos ativos na MEO e tipicamente monitorizam os sistemas, classificando as vulnerabilidades encontradas.

2.1 Princípios de um ataque

Os constantes ataques informáticos têm-se tornado um fenómeno comum na rotina das organizações. A presente dependência tecnológica motiva a exploração de novas abordagens criminosas, que estão em constante evolução. Esta criminalidade tem vindo a tornar-se cada vez mais prejudicial, sendo extremamente importante tomar medidas para tentar condicioná-la ao máximo.

Dia após dia são descobertos novos métodos de ataque que quebram os mais diversos sistemas. Estes métodos derivam do estudo da tecnologia e como esta pode ser manipulada, podendo ser altamente destruidores para as entidades que as empregam. A motivação para a procura e realização de ataques informáticos é diversa, no entanto, é possível classificar a maioria das motivações em alguns tópicos distintos como por exemplo:

- A curiosidade na perceção dos sistemas que suportam o serviço, e acidentalmente ou propositadamente, a exploração dos mesmos;
- Participação em programas de *bug bounty* suportados por empresas que pretendem fortalecer as suas infraestruturas, ou simplesmente *trophy hunt* para satisfação pessoal;
- Roubo de informação para uso pessoal ou venda;
- Sabotagem de sistemas por razões pessoais ou motivadas por interesses de outras entidades;
- Ciber guerra.

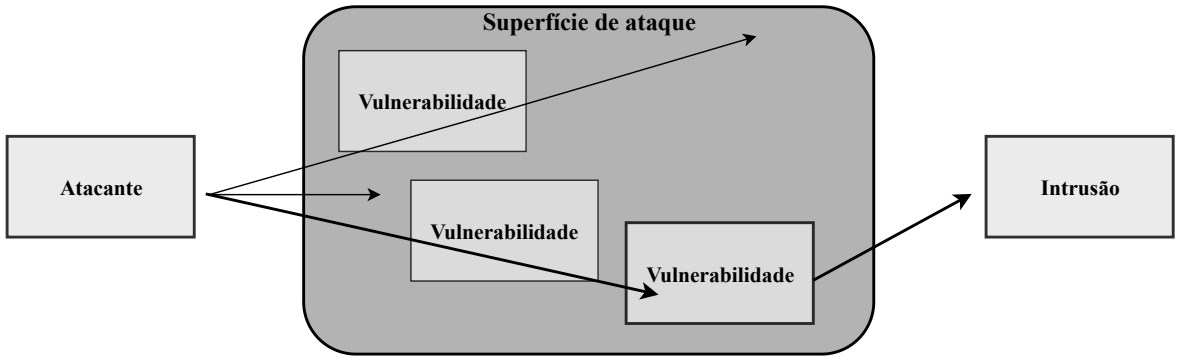


Figura 2.1: Modelo AVI

Para representar as etapas que tipicamente levam à intrusão, foi desenhado um modelo ilustrativo denominado por **Modelo AVI** (Ataque + Vulnerabilidade = Intrusão). Este modelo, representado na Figura 2.1, permite compreender o procedimento comum que provoca uma falha de segurança num sistema. O atacante tem como foco primeiro reconhecer o alvo, conseguindo extrair o máximo de informação do possível. Com a informação recolhida tenta-se confirmar a existência de vulnerabilidades conhecidas e com base nas vulnerabilidades existentes são elaborados métodos de ataque. Ao executarem os ataques, se estes forem bem sucedidos, é atingido o objetivo principal que é a intrusão no sistema. Após a intrusão é comum o estudo do funcionamento do sistema, para maximizar os ganhos do ataque consoante a motivação do mesmo.

2.2 Vulnerabilidade

> *"Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source."* *NIST*

Vulnerabilidade é um conceito para o qual não existe uma definição única. Deste modo, diversas organizações que operam na área da segurança informática criaram a sua própria definição. No âmbito deste projeto foi adotada a definição proposta pelo *National Institute of Standards and Technology (NIST)*, transcrita a cima. Sendo assim, uma vulnerabilidade é uma fragilidade num sistema de informação que pode ser explorada ou desencadeada por uma entidade maliciosa.

A exploração de uma vulnerabilidade geralmente origina a violação de pelo menos um dos três princípios fundamentais da segurança informática [7]: a confidencialidade da informação através do acesso não autorizado; a integridade de dados que são alterados ou eliminados por agentes maliciosos; e/ou a disponibilidade do sistema que é comprometida com falhas ou perturbações de funcionamento. Consequentemente, os ataques bem sucedidos provocam elevados prejuízos que podem não só comprometer informações sensíveis mas também, em casos extremos, descredibilizar organizações perante o público e investidores, como foi o caso de empresas como *Code Spaces, Telefonica, FlexiSpy e Medstar Health* [11].

2.2.1 Potencial impacto de uma vulnerabilidade

Cada vulnerabilidade é única e tem como base o sistema que é utilizado pela infraestrutura em causa, no entanto, a sua exploração pode provocar consequências distintas. De forma a desenvolver métodos

que explorem vulnerabilidades, é essencial compreender o funcionamento da vítima. Só através da sua compreensão é possível construir métodos que alterem o correto funcionamento, atingindo o fim desejado. De acordo com a *Intel* [17], existem três tipos de consequências causados pela exploração de uma vulnerabilidade, que são:

- **Escalar privilégios -** Um atacante ao explorar a vulnerabilidade é capaz de obter mais privilégios sobre um sistema comprometido, permitindo executar ações que normalmente não seriam possíveis;

- **Exfiltração de informação -** Um atacante ao explorar a vulnerabilidade consegue obter acesso a informação confidencial;

- **Negação de serviço -** Através da exploração da vulnerabilidade, é posto em causa o correto funcionamento do sistema, levando à sua falha.

Os três tipos de consequência sumarizam o impacto sobre as entidades comprometidas. Este impacto pode ser crucial no crescimento e expansão de uma empresa visto ser um processo extremamente trabalhoso, construído com base na reputação que é transmitida para o público. Por estes motivos, é compreensível que a melhor forma de evitar processos custosos de remediação de danos passa por investir na descoberta e prevenção da exploração das vulnerabilidades na infraestrutura. Esta deve ser a motivação que sustenta a análise contínua de sistemas.

Um dos maiores ciber ataques alguma vez feitos é conhecido como *WannaCry* [14]. Este foi reportado em maio de 2017 e teve impacto a nível mundial, comprometendo milhares de organizações. O ataque explorou duas vulnerabilidades críticas presentes nos sistemas, o *EternalBlue* [15] e *DoublePulsar* [16], permitindo a execução de código arbitrário em modo núcleo[1] e a implementação de um *backdoor*[2] na vítima. O ataque teve como objetivo a sabotagem através da cifra dos conteúdos armazenados, provocando enormes falhas na disponibilidade de serviço, perdas de informação e em alguns casos, pagamentos de elevadas quantias monetárias sem garantias de sucesso. Estas foram as consequências diretas do ataque, que causaram elevados prejuízos e que refletiram a falta de cultura de ciber higiene e monitorização de sistemas. Esta situação que deve servir como motivação para o nosso trabalho na área da gestão de vulnerabilidades.

2.2.2 Ciclo de vida das vulnerabilidades

O ciclo de vida de vulnerabilidades descreve as várias fases percorridas pelas vulnerabilidades desde a sua descoberta, até à mitigação e consequente resolução [8]. Este processo é dividido em quatro fases distintas que são executadas de forma sequencial, como representado na Figura 2.2:

[1] O modo núcleo é o componente central do sistema operativo.

[2] Um *backdoor* é um método que facilita o acesso ao sistema, escapando à autenticação e a outras proteções existentes.

Figura 2.2: Ciclo de vida das vulnerabilidades

Primeiro, ocorre a **descoberta** da vulnerabilidade, que pode acontecer em diversos contextos:

- As equipas de segurança da empresa fazem testes internos aos sistemas com o objetivo de perceber o quão vulnerável é a infraestrutura, e acabam por descobrir que existem vulnerabilidades previamente desconhecidas;
- Investigadores que estudam continuamente os mais diversos sistemas para não só melhorar os serviços fornecidos, mas também corrigir eventuais falhas de segurança que sejam encontradas;
- Agentes maliciosos que procuram descobrir vulnerabilidades que ainda não são conhecidas nem pelo público, nem pela própria organização, as chamadas *0-day vulnerabilities*. Estas vulnerabilidades são altamente valiosas, podendo ser facilmente comercializadas no mercado negro, uma vez que proporcionam vantagem sobre o alvo devido à inconsciência da suscetibilidade dos seus sistemas.

Nesta etapa, dependendo da entidade que descobre a vulnerabilidade, o desfecho pode ser diferente. A vulnerabilidade pode ser explorada até ser identificada, corrigida e posteriormente publicada, ou pode ser diretamente comunicada à entidade responsável pelo produto.

A **publicação** de vulnerabilidades é realizada sobre bases de dados mantidas por organizações padrão. Estas organizações pretendem gerir e armazenar de forma centralizada todas as publicações que foram efetuadas, facilitando a consulta pela comunidade. Como exemplos de bases de dados existem a *ISS X-Force* desenvolvida pela *Intel*, a *Security Focus* da *Symantec* e a *Common Vulnerabilities and Exposures (CVE)* mantida pela *MITRE* [20][19][2]. O processo de publicação leva algum tempo uma vez que é definido um período pré-publicação que é atribuído à empresa proprietária do sistema, para que idealmente consiga corrigir o problema identificado. No contexto deste projeto é considerada apenas a CVE devido à elevada importância e utilização pela indústria.

Com a publicação, tipicamente é desenvolvida e **lançada uma atualização**. Esta atualização vem aplicar as alterações necessárias para corrigir a vulnerabilidade sobre os sistemas que são afetados. De

modo a mitigar a existência da vulnerabilidade, habitualmente existe uma campanha de sensibilização para a instalação da atualização. A campanha geralmente decorre através da notificação dos clientes do produto, ou através de meios alternativos, como a disseminação em listas de mensagens relativas a tópicos de segurança. Quão mais rápido for a adesão à nova atualização, mais rápido se caminha para a mitigação da vulnerabilidade.

2.2.3 CVE

Common Vulnerability Exposures/Enumeration [2] ou **CVE** é uma base de dados que armazena descrições de vulnerabilidades publicamente conhecidas, gerida pela *MITRE*. Foi criada com os objetivos de oferecer um método de referência para consulta de vulnerabilidades na área da segurança informática. Esta base de dados ainda permite correlacionar informação disponível com outras bases de dados diferentes ou integração com novas aplicações em desenvolvimento.

A cada vulnerabilidade adicionada à lista são atribuídos três parâmetros:

- Um **CVE-ID** único de formato CVE-AAAA-ID, onde AAAA corresponde ao ano em que o CVE foi criado. Caso a vulnerabilidade seja pública antes da criação do CVE então AAAA corresponde ao ano que foi publicada. O ID corresponde ao identificador único da vulnerabilidade face ao ano em que foi criado o CVE.
- Uma breve **descrição** da vulnerabilidade, que contextualiza o leitor, informando a sua origem, possíveis consequências e limitações da sua exploração.
- Um conjunto de **referências** que contém informação adicional sobre a vulnerabilidade, tipicamente a versão de *software* afetada e publicações por parte dos fabricantes.

2.2.4 CVSS

Common Vulnerability Scoring System [3] ou **CVSS** é um sistema de classificação da severidade de vulnerabilidades criado pelo *National Infrastructure Advisory Council (NIAC)*. A classificação é representada de forma numérica e tem como base três grupos de métricas que quantificam o grau de severidade associado a cada tema. Esta permite obter uma perspetiva global do impacto da vulnerabilidade, possibilitando a comparação com outras e fornecendo uma base qualitativa da ameaça, bem como do estado dos sistemas.

Esta classificação é relevante pois está presente na grande maioria de ferramentas de segurança utilizadas, auxiliando a definição de prioridade na resolução. Sendo a MEO uma empresa de grande dimensão, são encontradas bastantes vulnerabilidades nos seus ativos, logo é necessário estabelecer uma ordem de resolução eficiente. Tipicamente para a resolução é considerado o rácio da facilidade da resolução face ao custo da resolução. No entanto, o CVSS permite adicionar mais um fator à equação, fator este que considera o impacto da ameaça associada. Quanto maior for o impacto causado pela exploração da vulnerabilidade, maior será o risco a que a organização está exposta, afetando diretamente a qualidade de serviço. Com a consideração de mais fatores na atribuição de prioridades de mitigação, este processo torna-se mais robusto e eficiente, propriedades que são fulcrais em grandes organizações.

O CVSS tem três vertentes que são calculadas através de grupos de métricas distintos, *i.e,* o grupo de métricas base, temporais e de ambiente [22].

Métricas base

Grupo de métricas que representa as características intrínsecas às vulnerabilidades, que não sofrem alterações ao longo do tempo nem com a mudança de ambiente. Estas métricas são compostas por submétricas que derivam de temas mais específicos, as métricas de exploração e de impacto, ilustradas na Figura 2.3.

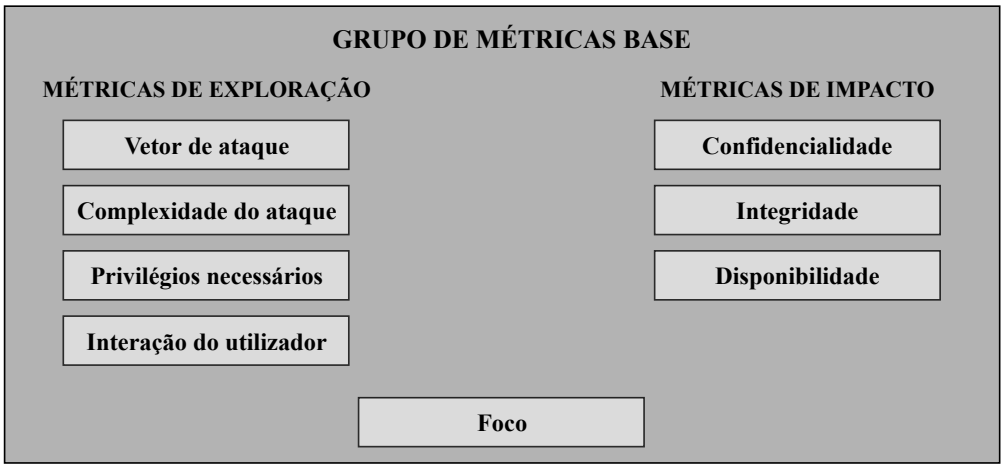


Figura 2.3: CVSS - Grupo de métricas base

As **métricas de exploração** consideram os vários fatores que condicionam a exploração de uma vulnerabilidade. Estes são: o vetor de ataque, que indica o contexto em que a vulnerabilidade existe; a complexidade do ataque, que considera as condições fora de controlo do atacante na exploração da vulnerabilidade; os privilégios necessários, baseados no nível de privilégios que o atacante tem de ter antes de explorar; e a vulnerabilidade e interação do utilizador, que tem em conta a necessidade de que um utilizador normal (que não o atacante), acabe por participar na exploração da fragilidade.

As **métricas de impacto** consideram os efeitos consequentes à exploração com sucesso da vulnerabilidade. Este impacto envolve os princípios da segurança informática, *i.e*, a confidencialidade, a integridade e a disponibilidade. O foco interage com as duas submétricas e considera o impacto de uma vulnerabilidade sobre o sistema, quando presente num componente.

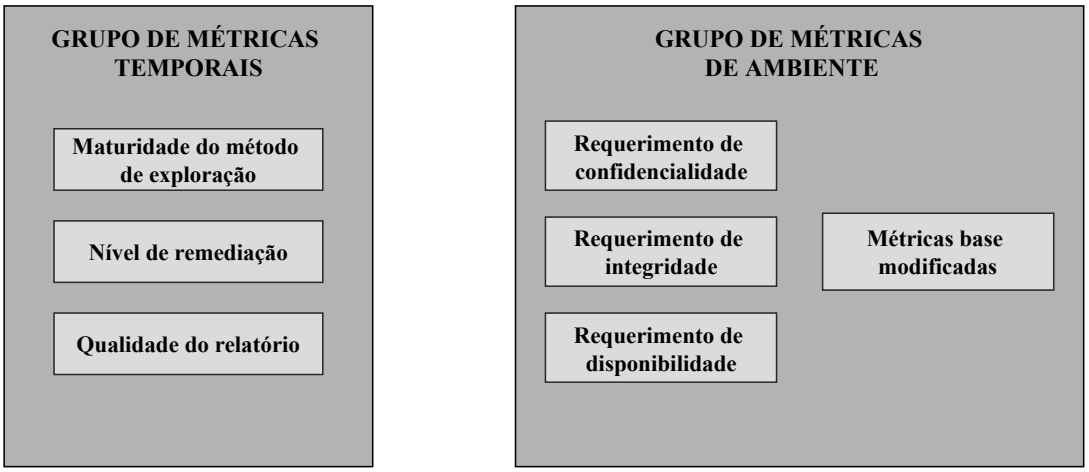


Figura 2.4: CVSS - Grupos de métricas temporais e de ambiente

Métricas temporais

O grupo de métricas temporais calcula o índice atual de ameaça devido à presença da vulnerabilidade nos sistemas. Este índice é baseado nos recursos disponíveis à data, representados na Figura 2.4: a **maturidade do método de exploração**, proveniente do código ou método de exploração que são conhecidos; o **nível de remediação**, baseado na existência de atualizações ou *workarounds* para o problema; e por fim a **qualidade do relatório** que considera a descrição disponível para a compreensão da fragilidade.

Métricas de ambiente

Este grupo de métricas permite ao analista configurar a pontuação CVSS de modo que esta fique de acordo com as condições únicas ao ambiente a avaliar. As métricas deste grupo estão representadas na Figura 2.4 e são divididas em dois subgrupos. O primeiro é referente aos requisitos de segurança, composto pela **confidencialidade**, **integridade** e **disponibilidade**. Estes requisitos devem ser definidos de acordo com a respetiva importância sobre os ativos em questão. O segundo grupo corresponde às **métricas base que foram modificadas** e permite ao analista sobrepor o valor das métricas base, de acordo com as características especificas do ambiente.

Para que seja possível categorizar os valores obtidos no cálculo da pontuação CVSS foi definido um padrão, que atribui graus de criticidade de acordo com os valores obtidos. Esta associação pontuação-classificação está representada na Tabela 2.1.

Classificação	CVSS Score
None	0.0
Baixo	0.1 - 3.9
Médio	4.0 - 6.9
Alto	7.0 - 8.9
Crítico	9.0 - 10.0

Tabela 2.1: Associação das classificações com a pontuação CVSS

2.3 Motores de deteção e análise de vulnerabilidades

Os motores de deteção e análise de vulnerabilidades (MDAV) são um elemento essencial no processo de gestão de vulnerabilidades, sendo concretizado pelas ferramentas que executam os varrimentos de vulnerabilidades existentes nos ativos da infraestrutura. Na Figura 2.5 pode-se observar a arquitetura genérica de uma ferramenta deste tipo.

As ferramentas contêm um motor de busca que aplica um conjunto de técnicas para confirmar a existência de fragilidades presentes no alvo. Cada MDAV é específico a uma área de incidência sobre o qual diferem as técnicas a aplicar, *i.e*, as técnicas utilizadas para analisar redes são diferentes das utilizadas para analisar máquinas ou aplicações *Web*. Estas técnicas de análise geralmente interagem com o serviço vulnerável a testar no alvo, tentando obter uma resposta que é posteriormente validada. Os procedimentos para os testes são geralmente armazenados em bases de dados que alimentam os motores de busca.

Com a obtenção da resposta é realizada a respetiva avaliação para conseguir processar os resultados, onde tipicamente é deduzido se um sistema se encontra vulnerável ou não através da sua resposta.

Estas fragilidades podem ser vulnerabilidades mas também podem ser más práticas aplicadas durante o processo de configuração, *software* desatualizado, utilização de algoritmos descontinuados, entre outros.

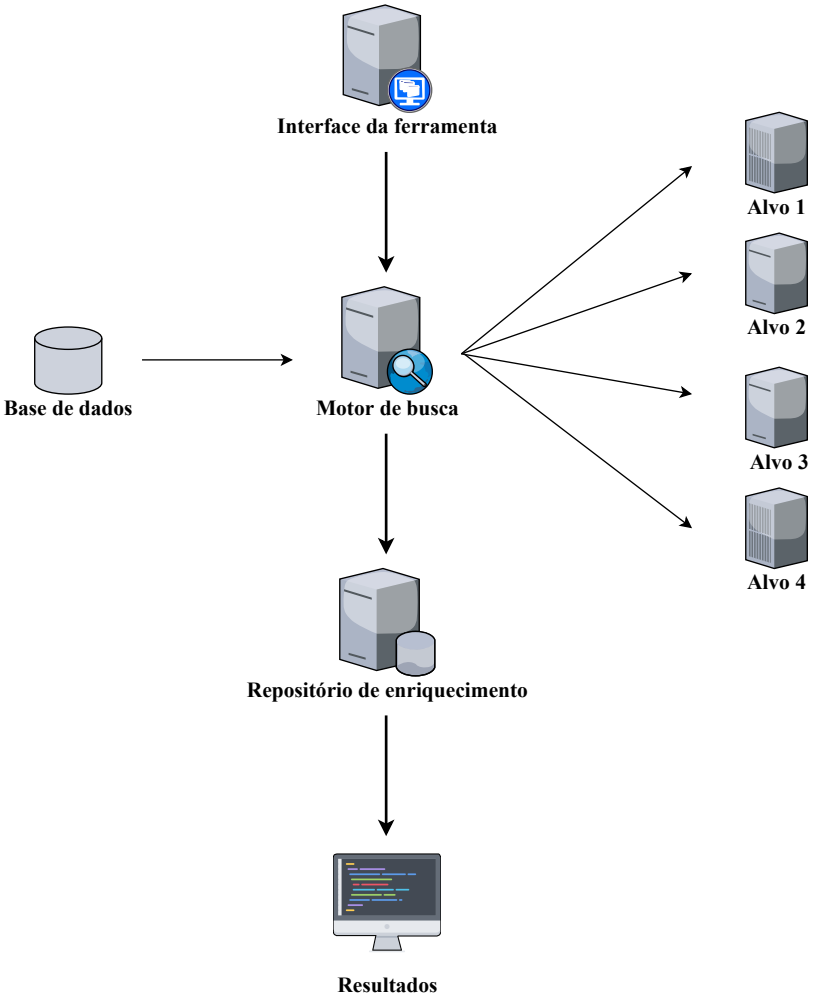


Figura 2.5: Arquitetura de um MDAV

Estas ferramentas são geralmente auxiliadas por um repositório de enriquecimento próprio que detém o objetivo de armazenar informação acerca das vulnerabilidades conhecidas (incluindo a facilidade de exploração, os índices próprios de criticidade e os métodos de exploração). Portanto, após serem identificadas começa o processo de cruzamento de dados para que se consiga apresentar a informação mais objetiva possível. Para cruzar dados é feita uma pesquisa de acordo com a vulnerabilidade encontrada no repositório e com a informação encontrada são enriquecidos os resultados.

Existem várias soluções de MDAV para diversos fins, no entanto, duas das propriedades que distinguem as ferramentas são o índice de falsos positivos e a qualidade da informação armazenada no repositório. O índice de falsos positivos reflete a precisão das técnicas de teste (devendo ser tão baixo quanto possível) e a informação presente no repositório de enriquecimento permite complementar de forma mais sólida os resultados produzidos, aumentando a produtividade de resolução.

2.3.1 Superfícies de análise

Ao utilizar um MDAV para detetar vulnerabilidades existem três tipos de varrimentos que podem ser aplicados [6]:

- **Varrimentos de vulnerabilidades internas**: Varrimento realizado sobre sistemas dentro da rede, excluindo os mecanismos de defesa externos, para compreender que vulnerabilidades podem ser exploradas, não só por agentes maliciosos internos à empresa que têm acesso privilegiado a funcionalidades restritas, mas também por agentes maliciosos que tenham ultrapassado as defesas de perímetro exteriores.

- **Varrimentos de vulnerabilidades externas**: Varrimento realizado de fora da rede da organização, com objetivo principal de detetar vulnerabilidades nas defesas de perímetro, como por exemplo, em portos abertos ou em anteparas.

- **Varrimentos de vulnerabilidades com autenticação**: Uma vertente que pode abranger os ativos internos e externos utilizando a autenticação para analisar com privilégios. São utilizadas credenciais de contas com permissões e é feita a análise dos sistemas do ponto de vista do utilizador. Este tipo de varrimentos é extremamente útil para ter uma perspetiva do nível de segurança contra-ataques internos (de utilizadores permitidos) e/ou agentes maliciosos que tenham conseguido permissões no sistema.

Tipicamente é possível configurar os varrimentos a executar, conseguindo-se personalizar por exemplo: os ativos a avaliar simultaneamente, o tipo de varrimento a realizar e o número de pacotes enviados. É importante que a configuração seja cuidadosamente executada de acordo com o alvo específico, para obter bons níveis de desempenho sem exceder as limitações dos ativos. Dado a interação direta dos MDAV com os ativos, iterando sobre as várias técnicas que dispõe de análise, é normal que alguns ativos que suportam serviços críticos tenham que ser individualmente considerados. Também devem ser tidos em conta as anteparas ou sistemas de deteção de intrusões (SDI) que podem bloquear o seu funcionamento, evitando desperdício de recursos que acabam por não ser produtivos. Não é de todo pretendido que a estabilidade de serviço seja comprometida pela análise destas ferramentas.

2.3.2 Área de incidência

A área de incidência de um MDAV corresponde à tecnologia sobre a qual vão ser focados os testes para encontrar vulnerabilidades. Este fator é relevante na escolha da ferramenta para analisar diferentes ambientes, dado que tipicamente cada ferramenta é exclusiva à análise de um ambiente especifico [1]. Em ambientes de rede existem:

- **Motores de análise de servidores *Web*** – Acedem a informação dos servidores *Web* conseguindo relacionar possíveis vulnerabilidades, por exemplo, através de ficheiros que podem comprometer o sistema.

- **Motores de análise de aplicações *Web*** – Fazem verificações sobre as aplicações *Web* implementadas em servidores para perceber o quão vulneráveis estas são a ataques comuns. Geralmente são

utilizados métodos que testam diversos *inputs* conhecidos como potenciais fontes de exploração de vulnerabilidades sobre campos que permitem entrada de informação, por exemplo: *Cross-site Scripting (XSS)* ou *SQL Injection (SQLi)*.

- **Motores de análise de rede** – Ferramentas com uso mais amplo, que combinam um conjunto de funcionalidades numa só solução, permitindo obter informações sobre vulnerabilidades no sistema combinando duas ou mais categorias das acima descritas.

2.3.3 Processo de funcionamento

O processo de funcionamento dos MDAV é dividido em quatro fases essenciais [5], como representado na Figura 2.6.

Figura 2.6: Processo de funcionamento de um MDAV

Reconhecimento

A primeira fase consiste no reconhecimento do alvo que se pretende testar. Esta fase difere de acordo com a área de incidência da ferramenta e é divida em dois passos que têm o objetivo de sondar o alvo para recolher o máximo de informação acerca das suas características e serviços. A primeira informação a recolher são os possíveis vetores de ataque presentes no alvo, *i.e,* em aplicações *Web* analisam-se os certificados testando a validade, os algoritmos criptográficos utilizados, entre outros, enquanto na análise de redes procuram-se portos abertos e protocolos associados (*User Datagram Protocol (UDP)* e *Transmission Control Protocol (TCP)*). Com a informação acerca dos serviços, são aplicados procedimentos para verificar que tecnologias são utilizadas, sabendo-se à partida que quanto mais informação existir mais fácil será a identificação de vulnerabilidades.

Enumeração

Após a identificação dos elementos, inicia-se a fase da enumeração, onde são organizados os elementos obtidos e é procurada mais informação específica sobre a infraestrutura (estrutura da rede, política de acessos, privilégios, campos de entrada de dados, etc...) de forma a consolidar a informação recolhida.

Descoberta de vulnerabilidades

Uma enumeração mais detalhada e rica torna mais fácil a descoberta de vulnerabilidades. Ao tirar partido da informação organizada sobre do alvo, é possível simplificar o processo de análise, resultando na atribuição de testes específicos de acordo com as propriedades obtidas na fase anterior.

Os testes aplicados geram indícios de exposição a determinadas ameaças que são depois comparadas com um inventário de casos conhecidos (geralmente armazenados em bases de dados ou na *cloud*). A

comparação de semelhança visa minimizar o número de falsos positivos, obtendo maior confiança sobre os resultados e evitando a identificação de vulnerabilidades que na prática não existem.

Geração de relatórios

Para finalizar, e para conseguir transmitir a informação de forma clara, são gerados relatórios que consolidam a informação relevante, de acordo com o público a que se destinam. Isto pois, relatórios a entregar a executivos têm âmbitos diferentes de outros que serão utilizados na resolução de problemas. No primeiro caso tipicamente é reportado o estado geral da infraestrutura, com dados estatísticos, enquanto no segundo é feita a descrição dos problemas encontrados. Na Figura 2.7 é possível observar um excerto de relatório executivo exemplo, produzido por um dos MDAV.

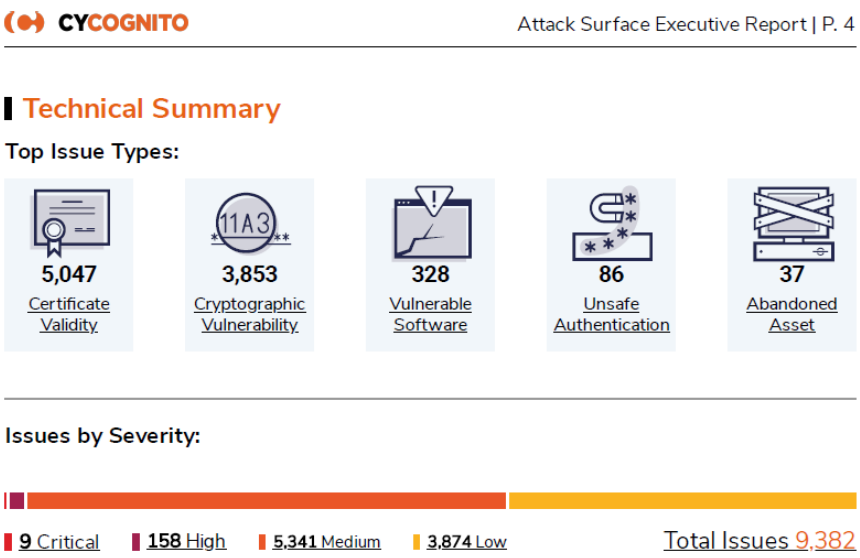


Figura 2.7: Excerto de relatório executivo produzido pelo Cycognito

2.3.4 Correlação da informação de várias ferramentas

Como descrito anteriormente, diferentes ferramentas funcionam sobre diferentes ambientes tecnológicos, logo identificam diferentes tipos de vulnerabilidades. No caso de empresas de telecomunicações, como é o caso da MEO, existe uma grande heterogeneidade de ambientes sobre os quais são implementados os diversos serviços fornecidos, com milhares de nós organizados em diferentes arquiteturas. Estes fatores tornam a escolha das ferramentas a utilizar uma tarefa complexa, uma vez que têm de ser estudadas as características dos sistemas para definir as ferramentas mais benéficas. Cada ferramenta pode ou não requerer um esforço financeiro, para a sua aquisição, mas todas requerem o investimento de recursos humanos para a sua aprendizagem e utilização, não sendo eficiente treinar pessoas para usarem uma ferramenta que não é importante no contexto da organização. Mesmo com uma boa avaliação de requisitos e com uma adequada escolha das ferramentas a configurar, não é possível encontrar ferramentas que produzam resultados livres de erros. Todas as ferramentas estão sujeitas à geração de falsos positivos com a produção de resultados errados, que têm de ser considerados durante a análise.

A estratégia adotada pela MEO tem o objetivo principal de recolher resultados o mais precisos

possíveis. Esta recolha passa por cruzar a informação das diversas ferramentas e com isto, aumentar o índice de confiança sobre a existência (ou não) das vulnerabilidades relatadas. O índice de confiança é sempre calculado de acordo com a reputação da ferramenta a utilizar, em que um MDAV mais conceituado tem um peso superior na equação em relação a um MDAV mais recente no mercado. Com base nesta confiança calculada, se duas ferramentas distintas reportarem a mesma vulnerabilidade é mais provável que esta realmente exista, logo é garantida a necessidade de resolução.

Contudo, para além de todos os benefícios associados à correlação de ferramentas, existem algumas condições que podem ter impacto na organização. Grande parte das ferramentas utilizadas para este propósito não são *open source* logo têm custos de aquisição ou subscrição. A quantidade de informação produzida cresce com o aumento do número de ferramentas utilizadas e esta informação tem de ser gerida, implicando investimento de recursos. O aumento de informação sobre um ativo, que em último caso pode ser um falso positivo (vulnerabilidades detetadas que efetivamente não existem) requer o processo de análise por um analista. Os analistas têm uma determinada capacidade de análise e reparação de vulnerabilidades, porém, esta decresce com o aumento de informação dispersa. Logo, a adoção desta estratégia requer uma análise prévia para garantir os melhores índices de custo/benefício. Esta análise baseia-se na identificação dos ativos críticos e na identificação das ferramentas que produzem melhores resultados sobre o ambiente desses ativos.

2.4 Ferramentas utilizadas

Um dos objetivos que se pretende atingir com a concretização deste projeto é o processamento automático dos relatórios obtidos. Esta automatização apenas é possível com a compreensão das ferramentas e o seu método de análise, sendo que, diferentes ferramentas produzem relatórios com estruturas e conteúdos diferentes. Nesta secção são descritos e exemplificados os vários MDAV's utilizados na MEO, bem como os resultados obtidos de cada um deles.

2.4.1 Cycognito

Cycognito é uma plataforma de testes de segurança baseada na nuvem, que simula automaticamente técnicas de intrusão e exploração de vulnerabilidades utilizadas por atacantes. Os objetivos desta plataforma são a pesquisa e deteção de ativos expostos à Internet e a respetiva avaliação do índice de segurança.

A plataforma é alimentada com os domínios a analisar e utiliza estes dados para aplicar técnicas de reconhecimento. Sobre as eventuais descobertas o processo repete-se, mapeando ao máximo a superfície de ataque da organização. Com os ativos identificados é realizada uma análise caso a caso, permitindo descobrir ativos que tenham sido esquecidos e também as vulnerabilidades presentes na infraestrutura. No caso da MEO, a utilização deste MDAV tira partido das propriedades de reconhecimento sobre ativos externos. Esta ferramenta classifica o grau de vulnerabilidade de acordo com:

- Ameaça em potencial;
- Complexidade da exploração;
- Complexidade de deteção;

- Esforço de remediação;
- Potencial impacto.

Esta classificação ajuda a definir uma ordem de prioridade de resolução, facilitando o trabalho da organização e consequentemente poupando tempo. Para exemplificar os resultados que são extraídos, pode ser consultado parte do relatório presente na Figura 2.8.

	A	B	C	D	E	F	G	H	I
1	Severity	Title	Organizatic	Environment	Platforms	Discoverability	First Seen	Last Seen	Issue Type
2	high	Vulnerable Apache Server Version		DBs		high	2019-11-02T	2019-11-08	Vulnerable Software
3	high	Vulnerability to ROBOT attack		Load Balancers		high	2019-05-20T	2019-11-08	Cryptographic Vulnerability
4	high	Vulnerable Apache Server Version		Other		high	2019-05-20T	2019-11-08	Vulnerable Software
5	high	Plaintext Authentication		Other		high	2019-05-20T	2019-11-08	Unsafe Authentication
6	high	Password Bruteforce Allowed		Other		high	2019-05-20T	2019-11-08	Unsafe Authentication
7	high	Vulnerable PHP Framework (5.3.3)		Other		moderate	2019-10-09T	2019-11-08	Vulnerable Software
8	high	Vulnerable ProFTBD 1.3.1		Other		high	2019-05-20T	2019-11-08	Vulnerable Software
9	high	Vulnerable Microsoft IIS 6.0 Server		Abandoned		high	2019-05-20T	2019-11-08	Vulnerable Software
10	high	Plaintext Authentication		Other		high	2019-05-20T	2019-11-08	Unsafe Authentication
11	high	Vulnerable Apache Server Version		DBs, Remote De		high	2019-05-20T	2019-11-08	Vulnerable Software
12	high	Vulnerable PHP Framework (5.3.3)		Other		high	2019-05-20T	2019-11-08	Vulnerable Software
13	high	Vulnerable Microsoft IIS 6.0 Server		Other		high	2019-05-20T	2019-11-08	Vulnerable Software
14	high	RC2 Cipher Support		Other		high	2019-05-20T	2019-11-08	Cryptographic Vulnerability
15	high	FREAK (Factoring RSA Export Keys)		Other		high	2019-05-20T	2019-11-08	Cryptographic Vulnerability
16	high	Exposed PHPInfo File		Remote Deskto		low	2019-05-19T	2019-08-18	Information Gathering
17	high	Vulnerable Microsoft IIS 6.0 Server		Other		moderate	2019-07-15T	2019-11-08	Vulnerable Software
18	high	Vulnerable Apache Server Version		Abandoned		high	2019-05-20T	2019-11-08	Vulnerable Software
19	high	Plaintext Authentication		Abandoned		high	2019-05-20T	2019-11-08	Unsafe Authentication
20	high	Vulnerable Apache Server Version		Abandoned		moderate	2019-05-20T	2019-11-08	Vulnerable Software

Figura 2.8: Exemplo de relatório técnico extraído da plataforma Cycognito.

2.4.2 Qualys

Qualys é uma plataforma de deteção de vulnerabilidades sofisticada, especializada em fornecer *Security as a Service* (SaaS), que oferece múltiplos serviços através da nuvem. Esta ferramenta permite agregar grupos de ativos através de gamas de endereços, catalogar tipos de varrimentos a executar com base em regras e definir estruturas de relatórios com base no seu propósito. Ainda dispõe vários módulos que abrangem diferentes áreas tecnológicas a analisar, no entanto, no contexto do projeto apenas serão considerados os seguintes:

- O módulo de *vulnerability management* atua ao nível dos ativos (tanto internos como externos) da infraestrutura da empresa, permitindo analisar o quão seguros estão, efetuando testes e analisando o retorno para compreender se existe evidência de vulnerabilidades conhecidas. Os resultados recolhidos são comparados com o repositório de enriquecimento para disponibilizar mais informação relacionada com as fragilidades e assim facilitar a remediação das debilidades encontradas.
- O módulo de *web application* atua ao nível das aplicações Web presentes nos ativos da infraestrutura, permitindo rastrear e catalogá-las, mesmo as que não sejam conhecidas. Analisa progressivamente cada aplicação Web para detetar a existência de vulnerabilidades, conseguindo abranger grande parte do top 10 da *Open Web Application Security Project* (OWASP) [23]. Este módulo também dispõe um modo autenticado para dar uma perspetiva da segurança interna a utilizadores com permissões.

Para exemplificar o tipo de informação extraída desta ferramenta pode-se consultar a Figura 2.9. Esta Figura contém um excerto de um relatório, sendo possível observar algumas das vulnerabilidades detetadas e alguma da informação disponibilizada.

	A	B	C	D	E	F	G	H
1	IP	DNS	OS	QID	Title	Vuln Status	Severity	Port
2			Windows 2008	90882	Windows Remote Desktop Protocol Weak Encryption Method Allowed	Active	3	3389
3			Windows 2008	90882	Windows Remote Desktop Protocol Weak Encryption Method Allowed	Active	3	3389
4			Windows 2008	90882	Windows Remote Desktop Protocol Weak Encryption Method Allowed	Active	3	3389
5			Windows 2008	90882	Windows Remote Desktop Protocol Weak Encryption Method Allowed	Active	3	3389
6			Windows 2008	90882	Windows Remote Desktop Protocol Weak Encryption Method Allowed	Active	3	3389
7			Windows 2008	90882	Windows Remote Desktop Protocol Weak Encryption Method Allowed	Active	3	3389
8			Windows 2008	42366	SSLv3.0/TLSv1.0 Protocol Weak CBC Mode Server Side Vulnerability (BE	Active	3	443
9			Windows 2008	42366	SSLv3.0/TLSv1.0 Protocol Weak CBC Mode Server Side Vulnerability (BE	Active	3	443
10			Windows 2008	42366	SSLv3.0/TLSv1.0 Protocol Weak CBC Mode Server Side Vulnerability (BE	Active	3	443
11			Windows 2008	42366	SSLv3.0/TLSv1.0 Protocol Weak CBC Mode Server Side Vulnerability (BE	Active	3	443
12			Windows 2008	42366	SSLv3.0/TLSv1.0 Protocol Weak CBC Mode Server Side Vulnerability (BE	Active	3	443
13			Windows 2008	42366	SSLv3.0/TLSv1.0 Protocol Weak CBC Mode Server Side Vulnerability (BE	Active	3	443
14			Windows Vista ,	42366	SSLv3.0/TLSv1.0 Protocol Weak CBC Mode Server Side Vulnerability (BE	Active	3	443
15			Windows Vista ,	38657	Birthday attacks against TLS ciphers with 64bit block size vulnerability (	Active	3	443
16			Windows Vista ,	38657	Birthday attacks against TLS ciphers with 64bit block size vulnerability (	Active	3	443
17			Windows 2008	38657	Birthday attacks against TLS ciphers with 64bit block size vulnerability (	Active	3	3389
18			Windows 2008	38657	Birthday attacks against TLS ciphers with 64bit block size vulnerability (	Active	3	443
19			Windows 2008	38657	Birthday attacks against TLS ciphers with 64bit block size vulnerability (	Active	3	3389
20			Windows 2008	38657	Birthday attacks against TLS ciphers with 64bit block size vulnerability (	Active	3	443

Figura 2.9: Exemplo de relatório técnico extraído da plataforma Qualys.

2.4.3 Bitsight

Bitsight é a organização responsável pelos *BitSight Security Ratings*, gerados através da análise contínua dos sistemas de outras empresas, organizadas por categorias. A análise é realizada utilizando diversos sensores espalhados geograficamente, que estudam constantemente tráfego e informação disponível na Internet para detetar ações suspeitas. Com base nos resultados da análise é feita uma comparação com a indústria de forma a calcular um índice de segurança, permitindo às empresas de um determinado setor determinar como é que a sua segurança se compara com a concorrência.

A classificação é calculada através de índices, podendo ser consultada a Figura 2.10 que ilustra um excerto de relatório. Estes índices são:

- **Sistemas comprometidos** – Considera os indícios de infeção de *botnet*[3], propagação de *Sending and Posting Advertisement in Mass* (SPAM), servidores infetados com *software* malicioso e existência de comunicações não solicitadas. Para calcular este índice são equacionados registos de diversos *sniffers*[4], que se encontram espalhados em vários pontos geográficos e ao detetar tráfego dos endereços da organização criam alertas.

- **Comportamento dos utilizadores** – Com base no tráfego de transferência de ficheiros, utiliza o protocolo do *BitTorrent* para avaliar que tipo de utilização é efetuada. Também são consideradas as credenciais expostas, através da verificação contínua de repositórios de informação (públicos e privados) onde costumam ser publicados dados provenientes de exfiltrações.

- **Diligência** – Analisa os servidores e serviços disponibilizados, tendo em conta características técnicas como cabeçalhos, portos abertos, configurações e a cultura de aplicação das atualizações.

[3]Grupos de computadores conectados à Internet que são controlados por um agente externo.
[4]Sistemas de análise de pacotes que intercetam e registam tráfego na rede.

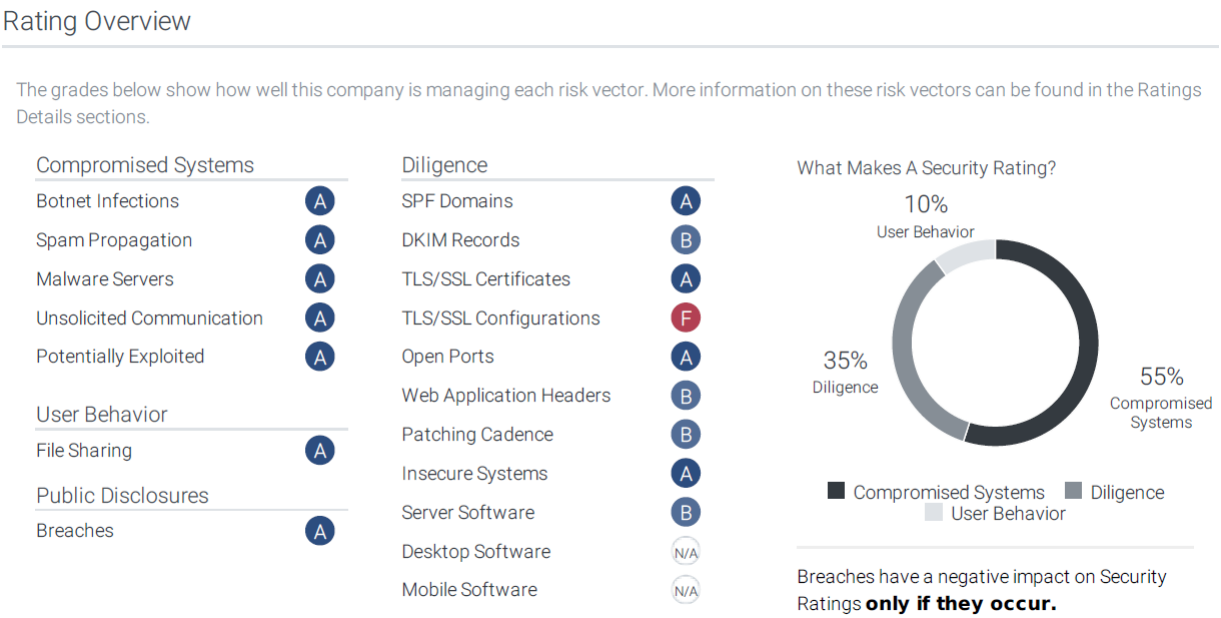


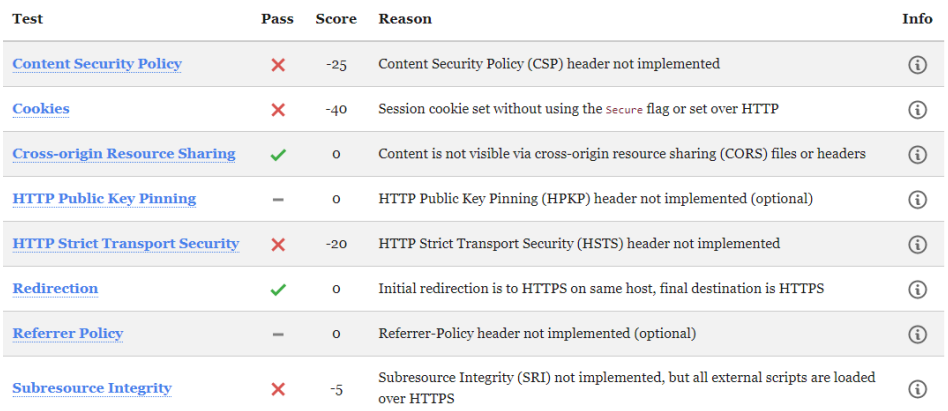
Figura 2.10: Excerto de relatório extraído da plataforma Bitsight.

2.4.4 Mozilla Observatory

Mozilla Observatory é um projeto *open source* disponibilizado pela *Mozilla*, que agrega um conjunto de análises para conseguir testar efetivamente as configurações e cabeçalhos de segurança das aplicações *Web*. Este projeto testa vários protocolos diferentes e executa também testes de aplicações de terceiros, no entanto, a MEO apenas utiliza resultados referentes ao *Hypertext Transfer Protocol (HTTP)*.

As verificações do protocolo HTTP têm o objetivo de classificar o alvo de acordo com as boas práticas e qualidade da respetiva configuração. São efetuados vários testes e a cada um é atribuída uma classificação, que serve como parâmetro na equação final para classificar o ativo. Na Figura 2.11 podem ser consultados alguns testes de configurações que são realizados.

Esta ferramenta analisa os pacotes trocados na comunicação com o alvo e através da informação disponível, averigua se há informação exposta que não deveria estar (tipicamente devido a más configurações). Também são analisados os cabeçalhos e as *cookies* do protocolo. Dos cabeçalhos são criados alertas se não forem adotadas as melhores práticas (cabeçalhos de segurança não implementados) e nas *cookies* é feita a inspeção das *flags* para dar uma perspetiva mais completa da configuração utilizada.



Test	Pass	Score	Reason	Info
Content Security Policy	✗	-25	Content Security Policy (CSP) header not implemented	ⓘ
Cookies	✗	-40	Session cookie set without using the Secure flag or set over HTTP	ⓘ
Cross-origin Resource Sharing	✓	0	Content is not visible via cross-origin resource sharing (CORS) files or headers	ⓘ
HTTP Public Key Pinning	–	0	HTTP Public Key Pinning (HPKP) header not implemented (optional)	ⓘ
HTTP Strict Transport Security	✗	-20	HTTP Strict Transport Security (HSTS) header not implemented	ⓘ
Redirection	✓	0	Initial redirection is to HTTPS on same host, final destination is HTTPS	ⓘ
Referrer Policy	–	0	Referrer-Policy header not implemented (optional)	ⓘ
Subresource Integrity	✗	-5	Subresource Integrity (SRI) not implemented, but all external scripts are loaded over HTTPS	ⓘ

Figura 2.11: Excerto de relatório extraído da plataforma *Mozilla Observatory*.

2.4.5 Processo de gestão de vulnerabilidades

O processo de gestão de vulnerabilidades é dividido fundamentalmente em três fases [10]: A **preparação**, o **varrimento de vulnerabilidades**, a **definição de ações de remediação**. Estas fases ocorrem de forma cíclica para garantir continuidade da gestão, como ilustrado na Figura 2.12.

Figura 2.12: Ciclo de funcionamento do processo de gestão de vulnerabilidades

Na fase de preparação é estudada a distribuição dos ativos da organização, identificando os que têm maior prioridade na monitorização (ativos críticos), bem como, que tecnologia utilizam. Com a distribuição dos ativos realizada, é possível definir o alvo de cada MDAV.

Se esta fase for corretamente desenvolvida são evitados atrasos na resolução de problemas encontrados e são reduzidos os longos períodos de espera motivados pela dimensão da análise. No caso da MEO, sendo uma empresa de grande escala, é fulcral preparar minuciosamente os ativos para a gestão, dado a sua ordem de grandeza. Em termos práticos e por uma questão de organização, é necessário dividir as várias máquinas em grupos de ativos que serão analisados em períodos diferentes, evitando níveis de concorrência que podem de alguma forma causar impacto no serviço prestado.

Para além da organização de ativos por área de incidência, é necessário definir corretamente qual o intervalo de tempo ótimo entre varrimentos. Este intervalo de tempo deve garantir que qualquer vulnerabilidade tornada pública após o último varrimento é detetada o mais rápido possível, respeitando as características dos ativos para não afetar a disponibilidade do serviço. No entanto, este processo pode fugir à regra, existindo sempre a possibilidade de executar varrimentos extraordinários se assim se justificar. Tipicamente os varrimentos não calendarizados decorrem com a publicação de vulnerabilidades criticas que se sabe existirem no *software* ou *hardware* utilizados, ou através da notícia de certos ataques. Daí a necessidade de analisar fora do previsto, para conseguir mitigar potenciais ameaças o mais rápido possível.

Logo, uma rotina de varrimentos ótima garante o maior número de execuções com o mínimo de intervalo temporal, diminuindo o período em que os sistemas estão vulneráveis a ameaças conhecidas, tal como é ilustrado na Figura 2.13.

Figura 2.13: Representação da janela temporal em que é desconhecida a vulnerabilidade dos ativos a uma ameaça já pública.

Com uma política de varrimentos bem definida, a probabilidade de se encontrarem novas vulnerabilidades é mais alta, uma vez que são feitos varrimentos de forma regular, com um período ótimo de intervalo. Tendencialmente, quanto mais rápido forem detetadas as vulnerabilidades, mais rápido serão efetivamente resolvidas.

Após o decorrer da primeira fase, são efetuados os varrimentos previamente configurados nas respetivas ferramentas de análise e são extraídos relatórios com os resultados encontrados. Os relatórios sumarizam de forma mais objetiva a informação obtida sobre os problemas encontrados, mencionando não só as vulnerabilidades do sistema mas também providenciando métodos de mitigação, qual o CVE correspondente para futura investigação e risco associado à permanência da mesma. Na fase de remediação devem-se tomar medidas para resolver potenciais problemas encontrados. No entanto, nem todas eles têm de ter solução, existindo três casos específicos de exceção que têm de ser considerados:

- Os **falsos positivos**, que são, as ameaças identificadas pelos MDAV que efetivamente não existem, *i.e.*, vulnerabilidades associadas a uma determinada versão, mas que no entanto foram corrigidas ou existe algum *workaround*;
- As vulnerabilidades com **risco moderado ou não críticas**, *i.e.*, vulnerabilidades que efetivamente estão presentes no sistema, mas que a sua solução implica a execução de medidas com elevado custo, que não justificam os recursos a despender;
- Vulnerabilidades que apenas **podem ser solucionadas mais tarde**, pois incidem sobre sistemas que no momento não podem ser alvo de alterações, devido ao impacto no negócio, ou vulnerabilidades que ainda não têm solução.

Após a identificação dos casos de exceção, resta um conjunto de vulnerabilidades que devem ser cautelosamente consideradas e corrigidas. Deste conjunto é ponderada qual a ordem de prioridade de resolução considerando o risco associado, facilidade no tratamento e tempo necessário investir.

Finalmente, são aplicadas as medidas de resolução sobre os problemas encontrados nos diversos sistemas e aguarda-se a execução do próximo varrimento que, não só, vem procurar novas vulnerabilidades que possam ter aparecido durante o intervalo de tempo decorrido, mas também, verificar se a solução implementada, que visou remover a vulnerabilidade, foi bem-sucedida. No próximo varrimento

são também gerados novos relatórios que têm de ser processados e é feita a correlação da informação com a previamente obtida.

Este processo decorre de forma cíclica, sendo um processo contínuo e de extrema importância para a correta manutenção de ativos e controlo de vulnerabilidades das organizações.

2.4.6 Projetos relacionados

No âmbito deste tema e também no contexto da MEO, foram desenvolvidos dois projetos de tese com propósito relacionado. Estes projetos deram origem ao *Vulnerability Assessment Coordinator* (*VAC*) e ao *Vulnerability Assessment Coordinator v2* (VACv2) [5][21].

O VAC foi resultado de um projeto de tese que dinamizou o processo de gestão de vulnerabilidades aplicado à data, utilizando OpenVAS e Nexpose[5]. A sua utilização permite a gestão dos MDAV de forma centralizada, conseguindo configurar e interagir com as ferramentas de deteção de vulnerabilidades num local único, sem a necessidade de interagir diretamente com cada ferramenta. Também tornou possível a calendarização de varrimentos, realização de análises espontâneas e posterior integração dos dados obtidos em bases de dados. Estes dados seriam depois disponibilizados num *Security Information and Event Management* (SIEM), tornando mais prático o processo de consulta e interação com as ferramentas, economizando recursos. No entanto, este projeto foi desenvolvido sobre os casos de uso da época, não deixando em aberto a possibilidade de alteração de componentes no futuro. Esta inadaptabilidade face às mudanças no processo de gestão de vulnerabilidades levou o projeto a ser posto de parte.

O VACv2 surge como uma nova abordagem ao tema da gestão de vulnerabilidades e que essencialmente vem melhorar o projeto VAC. O processo de melhoramento foi baseado na identificação e otimização dos pontos fracos do primeiro projeto:

- Dificuldade no uso da ferramenta;
- Difícil inclusão de diferentes tecnologias;
- Melhoria na extração de resultados.

Com estes tópicos bem definidos, foram propostos novos métodos de abordagem para o tornar mais consistente. Foi revisto o código e a estrutura base do VAC para aplicar melhorias, obtendo otimização por consequência. Melhoraram a interface existente, facilitando a sua compreensão e utilização. Foi re-implementado o sistema de integração de ferramentas, permitindo a adição de outras novas. Esta foi a principal contribuição do projeto, permitindo escalabilidade e obtendo melhores resultados em relação à primeira versão. No entanto, esta solução também acabou por não ser adaptada devido à constante alteração nas componentes do processo. As alterações nas componentes provocaram instabilidade de funcionamento, derivado ao curto espaço de tempo de realização do projeto.

O projeto a desenvolver distingue-se dos trabalhos anteriores pois não se pretende a criação de um novo sistema de gestão de vulnerabilidades, mas sim, automatização e otimização do atual. Através a recolha automática de relatórios, de quaisquer ferramentas, são correlacionados os dados e posteriormente armazenados numa base de dados. Com a informação agregada torna-se mais fácil a respetiva análise e com isto espera-se melhorias no atual processo de monitorização de vulnerabilidades.

[5] Ambos o OpenVAS como o Nexpose são ferramentas de análise de vulnerabilidades.

2.5 Conclusão

Neste capítulo foram apresentados os conceitos considerados fundamentais para ser possível compreender o âmbito do projeto. Para além da contextualização sobre tópicos relacionados com a segurança informática, foram ainda apresentados outros projetos, anteriormente desenvolvidos na MEO, com objetivos relacionados. É de notar que não foram referenciados quaisquer outras soluções, externas à MEO, devido à falta de soluções que propõe a mesma finalidade.

No próximo capítulo será apresentada a estrutura definida para a implementação do CSVMS, bem como os motivos que fundamentaram as decisões tomadas.

Capítulo 3

Estrutura do CSVMS

Para identificar os requisitos do projeto estudaram-se tanto os procedimentos adotados pelos analistas do *Cyber Security Operations Center* (CSOC) na monitorização de ativos como os resultados obtidos em projetos anteriores que acabaram por não ser utilizados internamente. Foi possível compreender que o foco do trabalho deveria ser na automatização de tarefas repetitivas e na organização da informação, promovendo o crescimento da produtividade e da ciber higiene. Como resultado deste projeto surgiu a criação do *Cyber Security Vulnerability Management System* (CSVMS), um repositório de vulnerabilidades, que resulta da agregação dos diferentes dados extraídos de MDAV.

3.1 Requisitos do projeto

Dado o propósito do projeto, foi definido o conjunto de características que é necessário cumprir para tornar possível a criação da solução, também denominado por requisitos do sistema. Estes requisitos dependem diretamente da interação do sistema a criar com a atual metodologia de trabalho adotada pelos funcionários da MEO, bem como das plataformas e tecnologias utilizadas. Deste modo, como requisitos do sistema considerou-se:

- **A modularidade do sistema**, isto é, apesar da diversidade de MDAV's que o compõem, deve existir algum nível de independência entre as ferramentas atualmente utilizadas e o sistema de gestão de vulnerabilidades. Os MDAV's empregues num dado período podem vir a ser eliminados ou substituídos por outros, não podendo ser motivo de bloqueio ou interrupção do sistema num todo.

- A **escalabilidade do sistema**. Considerando que a dimensão da MEO é propicia à adoção de novas tecnologias, que se complementam para monitorizar com maior detalhe os ativos que suportam o serviço, deve ser possível aumentar o número de alvos a analisar sem ocorrerem quaisquer problemas.

- A **compatibilidade com novas ferramentas**. O sistema deve ser implementado de forma a ser tão compatível quanto possível com novas ferramentas, tornando fácil a adição de outras ferramentas e assim suportando a escalabilidade e a modularidade também necessárias.

- A **simplicidade de utilização**. É importante que o sistema seja simples e fácil de utilizar. A melhoria da produtividade da equipa de analistas é obtida com a simplificação das tarefas que

têm de realizar, logo é necessário desenvolver uma plataforma prática e intuitiva para os seus utilizadores.

- A **qualidade da informação**. A qualidade da informação disponibilizada é um dos fatores mais importantes para determinar a qualidade dos relatórios operacionais. É necessário maximizar a obtenção de resultados úteis à consulta dos técnicos, para evitar desperdícios de recursos na análise e/ou resolução de problemas inexistentes.

- Por fim, a **eficiência do sistema** também é um fator relevante para a solução, motivando a redução de tarefas internas desnecessárias e a rapidez na produção de resultados.

3.2 MDAV'S incluídos no projeto

A MEO utiliza diversos motores de deteção e análise de vulnerabilidades, que auxiliam os processos de controlo e gestão de ativos. Estas ferramentas são utilizadas não só sobre os ativos da própria organização, mas também, sobre ativos de clientes. Inicialmente o projeto ambicionava a utilização de uma vasta gama de motores de deteção e análise de vulnerabilidades, sendo que alguns dos quais já estavam implementados e em utilização internamente. Nos restantes casos estaria prevista a operacionalização no meio corporativo, que deveria ocorrer ao longo do ano, para que mais tarde se pudesse implementar no projeto. No entanto, por motivos não relacionados com o CSVMS, acabaram por não ser incluídos no mesmo.

As Tabelas 3.1, 3.2 e 3.3, permitem visualizar os MDAV's considerados durante todo o processo de desenvolvimento, representando também a sua importância para a organização e o grau de exposição dos ativos que analisam. A importância para a organização foi dividida em duas categorias, os considerados **primários** e os **secundários**. Esta classificação foi atribuída com base na maturidade, confiabilidade e qualidade da informação produzida, sendo que, ao grupo dos primários foram associadas as ferramentas mais relevantes nas categorias anteriormente descritas. Em relação ao grau de exposição dos ativos é importante estabelecer a separação dos ativos considerados **internos** ou pertencentes à intranet, isto é, se comunicam apenas com máquinas presentes na rede interna da organização (não estando expostos diretamente ao exterior), de ativos considerados **externos** (que estão expostos à *internet*, podendo comunicar com outras máquinas exteriores à organização). No entanto, dependendo do MDAV em questão, estes podem analisar ativos internos, externos ou híbridos. Esta separação é relevante porque ferramentas que analisam ativos com diferentes graus de exposição fazem a atribuição de graus de severidade de acordo com métricas diferentes, *i.e*, uma versão desatualizada de um *software* numa máquina interna, pode ser considerada fonte de elevado risco por um MDAV que apenas considera o ativo em si, não contabilizando o facto deste não estar exposto ao exterior. Outra ferramenta que apenas analisa ativos expostos pode atribuir igual severidade a um cabeçalho HTTP incorretamente configurado. Assim é necessário considerar o contexto em que a análise foi feita para ponderar o risco real.

Na Tabela 3.1 podem ser consultados os motores de deteção e análise de vulnerabilidades que foram implementados na solução final. Todos eles são considerados primários pelo seu reconhecimento de mercado e também por pertencerem ao atual processo de gestão de vulnerabilidades da empresa, demonstrando resultados bastante satisfatórios na deteção de vulnerabilidades. É de notar que embora

tenha sido incluído um único MDAV que analisa a gama interna de ativos, o *Qualys*, este é responsável pela cobertura da grande maioria de ativos da empresa, produzindo a informação considerada necessária para a gestão de vulnerabilidades internas. Os MDAV's que acabaram por não ser implementados foram separados em duas outras tabelas de acordo com o motivo da exclusão da integração.

A Tabela 3.2 tem presentes dois MDAV's que estão atualmente disponíveis para utilização na empresa, mas que foi decidido que não seriam incluídos no projeto. No caso específico do *Nmap* foi verificado que apenas era utilizado para analisar portos abertos, produzindo informação que acabava por ser redundante, pois já era incluída nos resultados de outros motores de deteção e análise de vulnerabilidades. Já o *Mozilla Observatory* apenas era utilizado de forma esporádica, aquando necessário verificar más configurações em *websites* específicos. Portanto, foi decidido que dadas as condições de utilização, não era relevante a criação de processos de extração e processamento dos dados, acabando também por não ser incluído.

Finalmente a Tabela 3.3 representa dois MDAV's que não estavam em funcionamento na MEO, mas que estaria planeada a respetiva implementação operacional na empresa. Estes foram ponderados de incluir no projeto à condição que seriam primeiro instalados e configurados na empresa, para que se conseguisse mais tarde produzir dados úteis. No entanto, por motivos alheios, não foram disponibilizados a tempo e por consequência, não foram incluídos no projeto.

MDAV	Importância	Ativos internos	Ativos Externos
Qualys	Primário	✓	✓
Bitsight	Primário	**X**	✓
Cycognito	Primário	**X**	✓

Tabela 3.1: Motores de deteção e analise de vulnerabilidades implementados no projeto.

MDAV	Importância	Ativos internos	Ativos Externos
Nmap	Secundário	✓	**X**
Mozilla Observatory	Secundário	**X**	✓

Tabela 3.2: Motores de deteção e analise de vulnerabilidades utilizados internamente mas que não foram implementados no projeto.

MDAV	Importância	Ativos internos	Ativos Externos
Harpoon	Primário	✓	✓
Shodan	Secundário	**X**	✓

Tabela 3.3: Motores de deteção e analise de vulnerabilidades que não chegaram a ser implementados internamente à MEO, não podendo ser usados no projeto.

3.3 Modelo do projeto

Como resultado do projeto a desenvolver surge o CSVMS, um sistema que pretende melhorar os índices de eficiência dos analistas de segurança da MEO, otimizando a capacidade de resolução de problemas num determinado espaço de tempo. O CSVMS, com a arquitetura ilustrada na Figura 3.1, pretende simplificar o método como as tarefas são atualmente realizadas tirando partido da automatização de diversos processos.

Numa primeira fase pretende-se criar um repositório central de vulnerabilidades, que é alimentado automaticamente, para que mais tarde se consiga efetuar consultas de forma simples, clara e eficaz. Para atingir este fim o sistema é responsável pela extração de relatórios a partir das fontes, processamento da informação extraída, armazenamento e representação de resultados. A concretização do projeto é baseada numa arquitetura constituída fundamentalmente por três módulos que se complementam para tornar o sistema funcional:

- **Motor de calendarização** - Responsável pela automatização da execução dos outros módulos de acordo com a frequência temporal definida.

- **Motor de processamento** - Responsável pela interação direta com os motores de deteção e análise de vulnerabilidades, extração dos respetivos relatórios, processamento de dados recolhidos e armazenamento normalizado.

- **Motor de visualização** - Responsável pela interação com a informação armazenada, processamento e pesquisa para que sejam visualizados resultados a ser consultados.

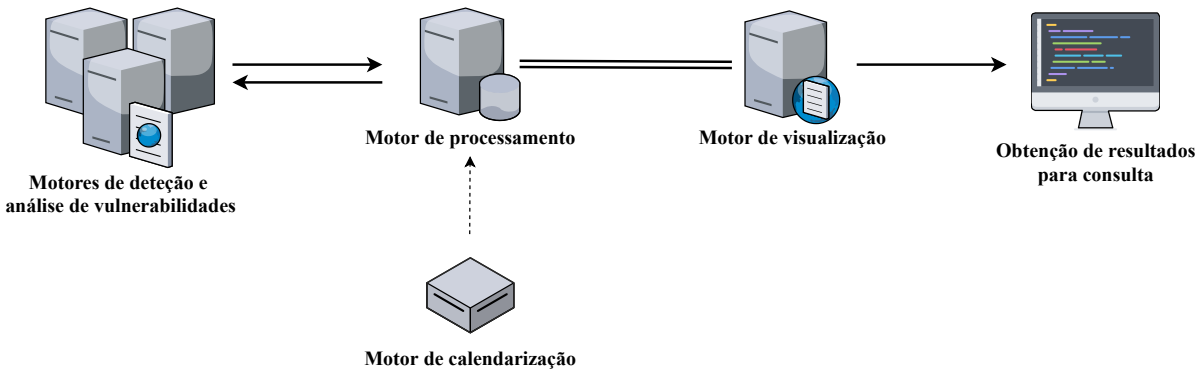


Figura 3.1: Arquitetura do CSVMS

3.3.1 Motor de calendarização

O módulo de calendarização (Figura 3.2) é responsável pela coordenação entre interações com diversos motores de deteção. Este atribui espaçamentos temporais adequados a cada MDAV, de forma a sincronizar da melhor forma possível a extração dos resultados, sem provocar problemas de escrita concorrente na tabela única. Através da frequência de análise e do relógio da máquina (tempo corrente) é calculada a *timestamp* que indica quando será realizada a próxima extração para cada motor de deteção e análise de vulnerabilidades.

Dado que nem todas as ferramentas têm datas específicas para publicação de resultados, como por exemplo o *Cycognito*, foi definido uma periodicidade semanal de recolha de informação. A extração realizada de semana a semana, assegura que o tempo de espera pelos novos resultados é no máximo sete dias, evitando recolhas múltiplas desnecessárias e permite que exista sempre informação para consulta. Ainda que não seja possível otimizar ao máximo os processos de extração com as publicações das fontes, ao seguir esta abordagem é reduzido o número de dias a aguardar de forma equilibrada.

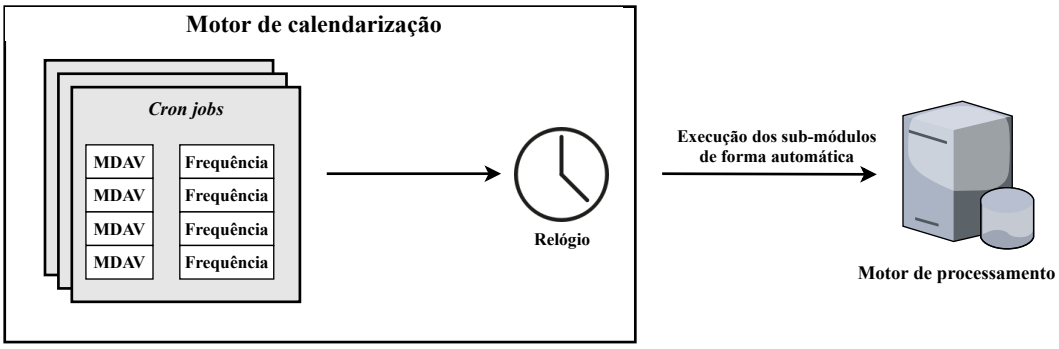


Figura 3.2: Motor de calendarização

3.3.2 Motor de processamento

A informação sobre eventuais ameaças presentes nos ativos monitorizados é um pré-requisito para a execução do processo da gestão de vulnerabilidades. Esta advém dos motores de deteção e análise de vulnerabilidades que são responsáveis por analisar e reportar quaisquer suspeitas de fragilidades encontradas. Para criar um sistema de gestão de vulnerabilidades prático e eficiente, foram divididas as várias tarefas necessárias a realizar em módulos, de maneira a torná-lo o mais modular possível.

O **motor de processamento** é o módulo do sistema que interage diretamente com cada MDAV para recolher e processar resultados de forma automática, tirando partido de componentes auxiliares que contêm o método de recolha para cada extração e o conhecimento para processar os dados. Já com os relatórios armazenados localmente é realizado o processamento da informação, normalizando-a e consequentemente promovendo a sua homogeneidade. Por fim esta componente é responsável por armazenar toda a informação já no formato canónico, produzindo o *input* necessário ao funcionamento do **motor de visualização**.

A Figura 3.3 ilustra as várias fases operacionais deste componente, que serão explicadas em detalhe mais à frente, e que resultam na produção de um relatório normalizado capaz de ser importado para a base de dados.

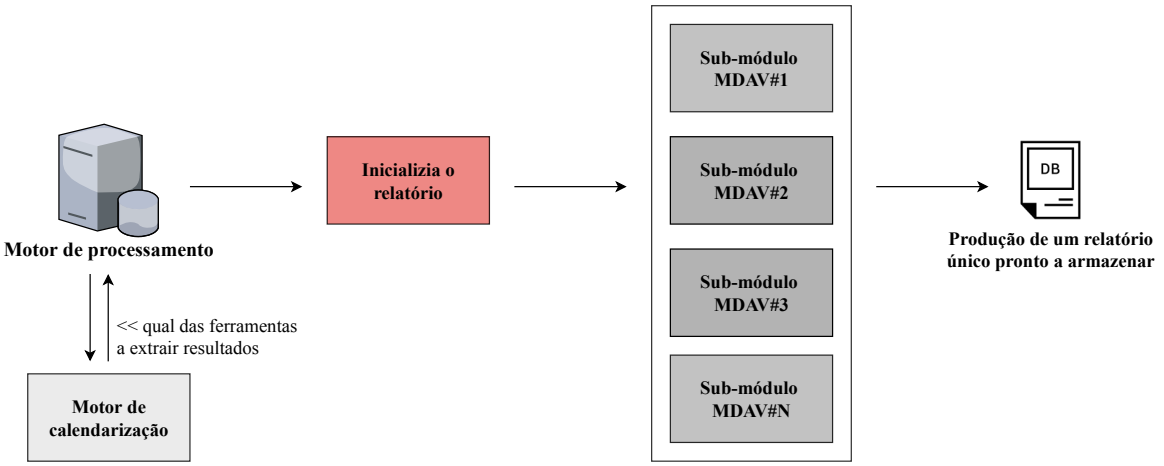


Figura 3.3: Arquitetura do motor de processamento

Inicialização do relatório de agregação

Dado o objetivo de centralizar a informação proveniente de diversas fontes, é necessário compreender corretamente cada uma delas para que se consiga de forma coerente e objetiva mapear corretamente o seu conteúdo. Cada MDAV categoriza a informação que recolhe com diferentes termos, métricas e referências na indústria, sendo necessário criar uma estrutura base de relatório que seja capaz de armazenar dados, independentemente da sua origem, identificando univocamente a informação.

Para tal, devem ser analisados os dados extraídos dos diferentes motores de deteção e análise de vulnerabilidades e com base nas categorias utilizadas, definir uma nomenclatura universal para a categorização do relatório único. Esta nomenclatura deve conter termos robustos e bem definidos, idealmente adaptada do MDAV com maior abrangência de ativos analisados. No entanto, devido à diversidade de informação recolhida de ferramenta para ferramenta, caso existam outras categorias relevantes, apenas associadas a dados de um determinado MDAV, estas podem também ser adicionadas à estrutura do relatório para torná-la o mais completa possível, como ilustrado na Figura 3.4.

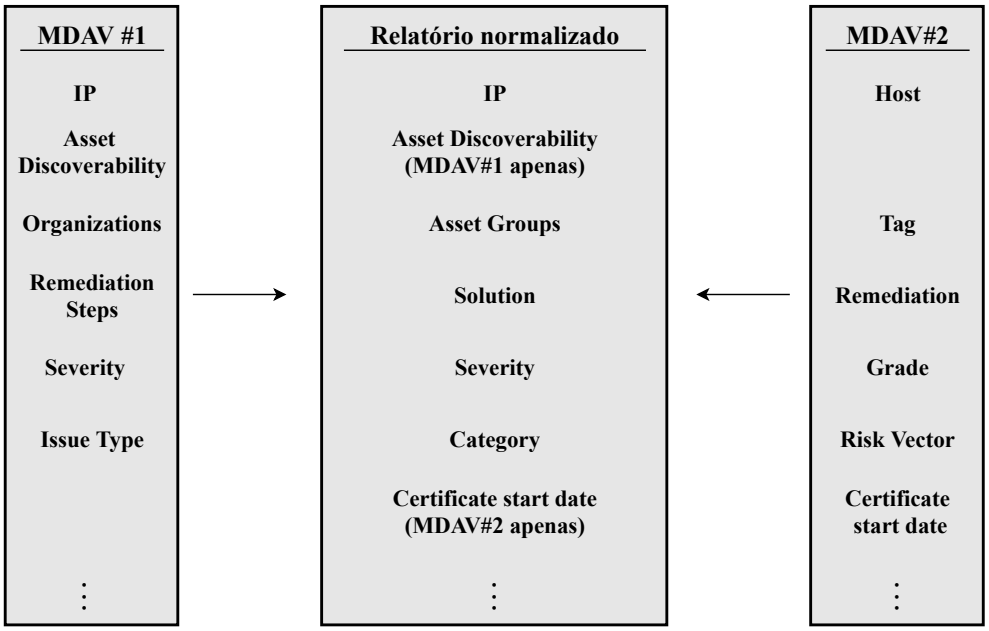


Figura 3.4: Relatório de agregação de informação

As categorias utilizadas serão o índice que guia os analistas durante o processo de consulta da informação, sendo portanto essencial que descriminem de forma abrangente os vários tópicos que podem auxiliar eficazmente o processo de resolução de vulnerabilidades. Para tal devem ser consideradas categorias com cobertura em diferentes âmbitos de consulta, como representado na Figura 3.5. Resumidamente, este relatório contém todas as categorias de todos os motores de deteção e análise de vulnerabilidades utilizando uma terminologia universal e servirá como base para efetuar o mapeamento da informação. É um elemento fundamental para o funcionamento do **motor de processamento**, inicializado antes da execução dos processos de extração e por fim servirá como *input* à base de dados.

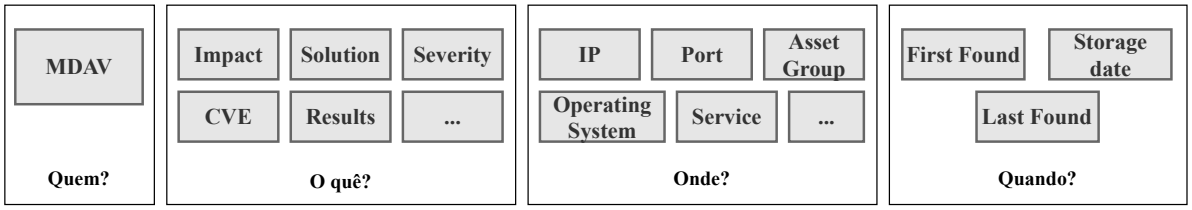


Figura 3.5: Relação entre as categorias de informação e o seu âmbito informativo

Sub-módulos de processamento

Como descrito anteriormente, dadas as diferenças das várias ferramentas que constituem o projeto e para cumprir o requisito de modularidade, é necessária a existência de um sub-módulo por motor de deteção e análise utilizado. Cada sub-módulo (Figura 3.6) serve como adaptador e meio de interação entre o **motor processamento** e o respetivo MDAV, contendo todo o conhecimento necessário para tornar possível a interação. Os sub-módulos são essencialmente constituídos por duas componentes de trabalho:

- **Componente de extração** - Contém todas as operações e informações necessárias para a correta execução e extração de resultados. Cada sub-módulo está associado a um único motor de deteção e análise de vulnerabilidades e quando é recebida a ordem de executação do **motor de calendarização**, este aplica o método de extração de relatórios. Por fim armazena todos os resultados temporariamente na máquina local.

- **Componente de processamento** - Contém todas as operações e informações necessárias para o correto processamento da informação previamente extraída e armazenada localmente. Usufruindo do relatório inicializado e de acordo com a estrutura do relatório extraído, será efetuada a normalização, enriquecimento e mapeamento da informação. Este componente efetua a grande maioria dos cálculos e conversões para assegurar a compatibilidade da estrutura dos dados com os restantes elementos do projeto.

A agregação das operações sobre MDAV's em sub-módulos permite não só gerir de forma mais eficaz qualquer alteração que seja necessária realizar sobre os mesmos, mas também assegura o funcionamento do sistema mesmo que nem todas as componentes estejam a funcionar corretamente. Esta metodologia acaba por ser útil tanto na adição de novas funcionalidades ao projeto, como para a identificação e correção de eventuais problemas ou erros que possam surgir numa dada ferramenta.

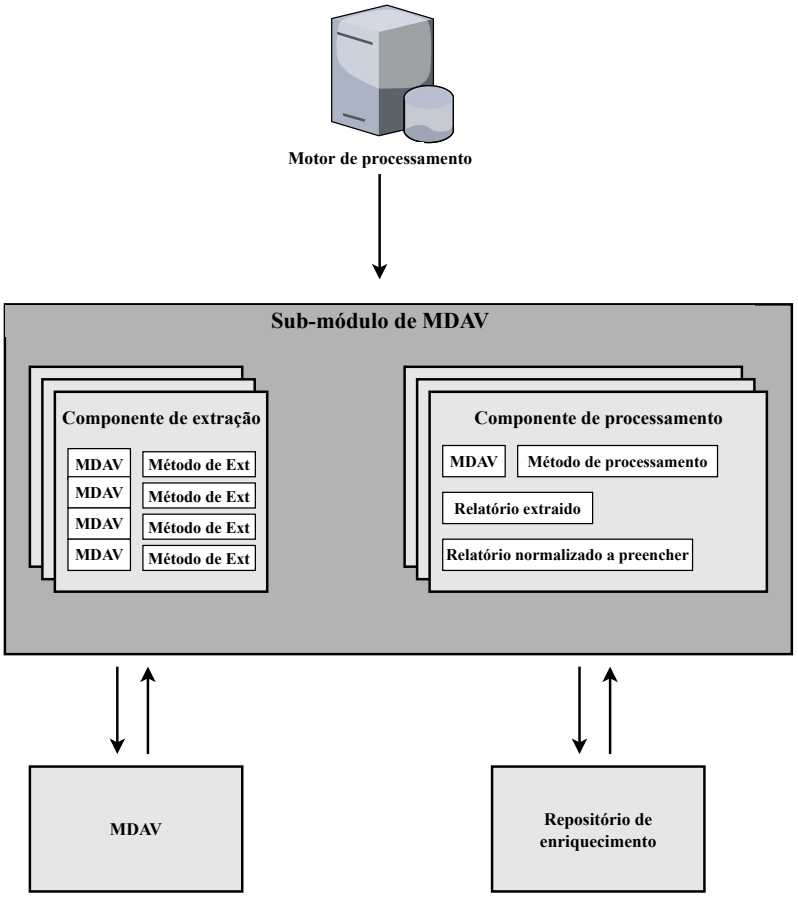


Figura 3.6: Estrutura do sub-módulo

Processo de extração de informação

O processo de extração de informação é a etapa que irá alimentar todo o projeto para que futuramente se consiga consultar dados essenciais ao trabalho dos analistas da MEO. Este processo resulta da interação direta do motor de processamento, através dos sub-módulos, com os vários motores de deteção e análise de vulnerabilidades. Cada motor de deteção e análise de vulnerabilidades é único, e estrutura os resultados produzidos num determinado formato de representação, tipicamente em *JavaScript Objet Notation* (JSON), *Extensible Markup Language* (XML) e *Comma-separated values* (CSV). Estes formatos podem variar de acordo com o método de extração utilizado, logo requerem um processamento específico caso a caso.

Nesta fase foi necessária a análise de cada ferramenta, para perceber quais as opções de extração existentes e também, qual seria a estrutura dos dados resultante de cada uma delas. Ainda que nem todos os motores de deteção e análise de vulnerabilidades se encontrem em níveis equiparáveis de desenvolvimento, o critério de escolha no mecanismo de extração foi sempre a simplicidade e eficiência, acabando por resultar em duas opções distintas:

- Preferencialmente a utilização da *Application Programming Interface* (API), caso esta esteja implementada e bem definida. Normalmente é mais estável, mais eficiente e tem suporte direto do produtor da ferramenta, podendo oferecer maior escalabilidade.
- A utilização de *Robotic Process Automation* (RPA) caso não exista uma API disponível. Esta

abordagem embora menos eficiente, permite programar robôs (processos que trabalham automaticamente) que replicam ações humanas, conseguindo extrair os relatórios do próprio *website*.

Na prática, este processo inicia-se sempre com a comunicação do motor de calendarização com o motor de processamento, que indica o momento de execução do processo de extração. Este é seguido pela execução dos sub-módulos que interagem com o respetivo MDAV para extrair a informação, armazenando-a localmente. Um exemplo desta interação está representado na Figura 3.7, um dos casos de uso genérico, composto por duas entidades com objetivos diferentes. Qualquer que seja o MDAV em utilização, o motor de processamento apenas é responsável pela obtenção de dados úteis ao projeto, sendo os processos de configuração da ferramenta e seleção de ativos a analisar responsabilidade da equipa de cibersegurança, que utiliza os painéis de controlo disponibilizados pelos próprios motores de deteção e análise de vulnerabilidades.

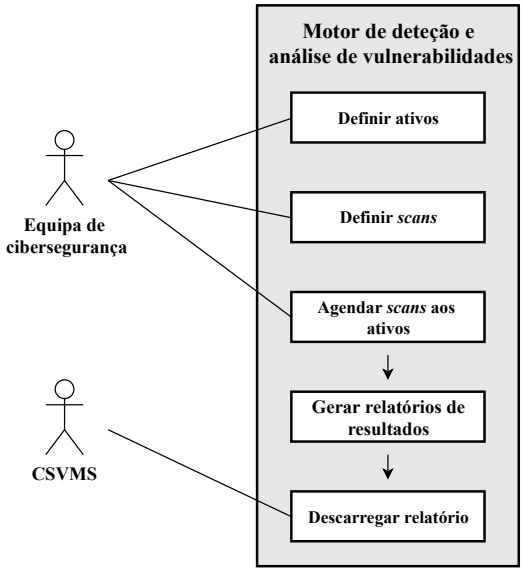


Figura 3.7: Caso de uso genérico do processo de extração de informação

Processamento da informação recolhida

Após o armazenamento local da informação recolhida pelos motores de deteção e análise de vulnerabilidades, os sub-módulos são responsáveis pelo processamento dos dados, normalizando e enriquecendo-os para que sejam agregados num único relatório, previamente inicializado.

Esta etapa é iniciada com a adoção de formatos universais para representar dados cuja interpretação será necessária mais tarde, nomeadamente para datas e valores numéricos. Este processamento deve-se às incoerências que podem ocorrer ao representar a mesma informação seguindo parâmetros diferentes. Para além da normalização de dados, ainda é enriquecida a informação existente sobre ativos em que foram detetadas vulnerabilidades. Para tal, é consultado um repositório auxiliar que contem informação adicional acerca do âmbito dos ativos na organização. Este repositório permite fazer a associação de cada ativo com o tipo de serviços prestados, a importância desses serviços para a empresa e também o grau de exposição dos mesmos. O objetivo é centralizar o máximo de informação disponível sobre os ativos vulneráveis, facilitando o processo de consulta e análise realizado pelos analistas da MEO.

Após o cruzamento da informação extraída com o repositório de enriquecimento, os sub-módulos começam o processo de mapeamento sobre a estrutura previamente definida, garantindo que toda a informação já processada, independentemente da fonte de origem, encontra-se armazenada num ficheiro único, como representado na Figura 3.8.

Também foi ponderada uma tentativa de correlação de informação, com o objetivo de identificar as entradas no relatório único que correspondessem a uma mesma vulnerabilidade. Esta possibilidade existe pois foram verificados casos em que diferentes motores de deteção e análise de vulnerabilidades analisam o mesmo ativo, potencialmente identificando em duplicado a mesma informação. Neste âmbito, foi experimentada uma análise dos dados utilizando processamento de linguagem natural, que acabou por não se verificar viável devido à insuficiência da informação disponível e à falta de consenso na utilização de métricas que permitissem a associação de informação. Assumiu-se por fim que cada vulnerabilidade reportada é considerada uma vulnerabilidade única pelo CSVMS, ficando à responsabilidade do analista a identificação manual da existência de duplicados.

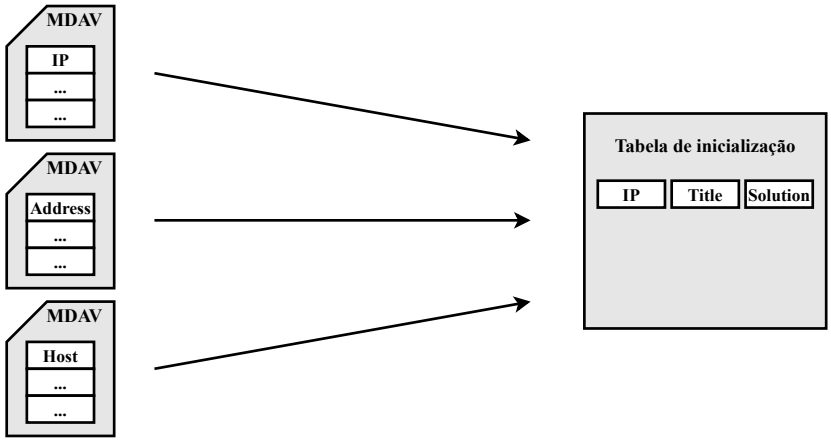


Figura 3.8: Centralização da informação previamente dispersa num só documento.

Cálculo do risco dos ativos

Como anteriormente referido, é necessária a existência de um índice que facilite a criação de uma ordem entre problemas a solucionar. A ordem de prioridade deverá ser estabelecida de acordo com a relação numérica entre o risco calculado para os diversos ativos, assumindo que, as vulnerabilidades detetadas num ativo cujo risco é mais elevado têm mais urgência de ser corrigidas em relação a outras detetadas em ativos cujo risco é menor. Este cálculo apenas é realizado após o mapeamento da informação de todos os motores de deteção e análise de vulnerabilidades, pois é nesta fase que é possível ter uma perceção geral acerca do número total de vulnerabilidades descobertas.

O cálculo do risco tem em conta dois fatores: o primeiro considera características do próprio ativo para o negócio, e o segundo foca-se na informação sobre vulnerabilidades presentes. Para cada variável da equação foram atribuídos pesos de acordo com a sua importância face à possível ameaça. Os fatores incluídos na fórmula de cálculo do risco são:

Informação sobre as características do ativo

- **Importância do serviço disponibilizado pelo ativo** - Foram definidos vários painéis que visam distinguir o nível de importância dos ativos da organização. O nível de importância varia de acordo com a necessidade dos serviços prestados para a continuidade de negócio, sendo atribuído a cada ativo um único painel. De acordo com o painel atribuído foi ponderado um peso para que seja contabilizado no momento de cálculo do risco.

- **Exposição do ativo** - A exposição de um ativo à *internet* (exposto ao exterior) ou apenas à *intranet* (exposto à rede interna) é uma das variáveis consideradas no cálculo. Assumimos que, qualquer ativo vulnerável e com ligação à *internet* é substancialmente mais crítico que um ativo apenas exposto à rede interna.

Informação sobre as vulnerabilidades do ativo

- **Média da severidade das vulnerabilidades detetadas**

  Cada vulnerabilidade detetada é classificada com uma severidade atribuída pelo motor de deteção que a reporta. Este valor é tão mais elevado quanto mais fácil for a sua exploração e também quão maior for o respetivo impacto. No cálculo do risco é incluída a média da severidade das vulnerabilidades existentes para que seja possível distinguir ativos com vulnerabilidades mais severas, de ativos com vulnerabilidades menos criticas.

- **Número de vulnerabilidades detetadas**

  É feita a contabilização das potenciais vulnerabilidades detetadas em cada ativo permitindo compreender quais os ativos com maior número de vulnerabilidades. Este valor é importante ser incluído na fórmula de cálculo pois permitirá atribuir maior risco a um ativo cuja média da severidade é elevada, com bastantes vulnerabilidades presentes, de um ativo que, por exemplo, contem apenas uma vulnerabilidade de alta severidade.

O cálculo do risco é dado por:

$$fator_{vulnerabilidades} = (0.2 * média\_severidades) + (0.8 * numero\_vulnerabilidades)$$

$$fator_{ativos} = ((0.65 * painel\_atribuído) + (0.35 * exposição\_atribuída))$$

$$\textbf{risco bruto} = 0.7 * fator_{ativos} + 0.3 * fator_{vulnerabilidades}$$

Cada variável é multiplicada por um valor que para além de representar a importância de determinada componente no cálculo, também reduz o valor final obtido para valores de menor dimensão. Após a obtenção do risco foi criada uma escala (Tabela 3.4) que permite agregar ativos de acordo com a categoria de risco em que se encontram. Considerando novamente que valores mais altos representam risco maior.

Risco bruto	Categoria de risco
0 - 19	1
20 - 39	2
40 - 59	3
60 - 79	4
80+	5

Tabela 3.4: Representação das categorias de risco de acordo com o risco bruto calculado.

Após serem calculados os valores do risco bruto e a respetiva categoria, ambos são armazenados no relatório juntamente com toda a informação acerca das vulnerabilidades, dando início ao processo de armazenamento da informação numa base de dados.

Armazenamento em base de dados

O processo de armazenamento é a última tarefa realizada pelo **motor de processamento** que pode ser também designado pela conclusão do processo de *Extract, Transform and Load* (*ETL*). Nesta etapa já todos os resultados independentemente das fontes foram extraídos, processados e convertidos num único artefacto pronto a ser armazenado num repositório. O repositório em questão irá conter a informação processada, tornando mais simples o processo de consulta e a monitorização das vulnerabilidades encontradas.

Para armazenar a informação, foram tidos em conta os diferentes tipos de bases de dados e a adaptabilidade de cada um deles, dado que a informação a armazenar é altamente sensível, com proporções tão grandes quanto maior for o número de ativos. Assim, foi necessário definir uma estratégia de armazenamento dependendo da fonte da informação, para permitir consultar e manipular dados quando necessário.

Com os dados armazenados numa base de dados segura, está concluído toda a execução do motor de processamento, sendo o **motor de visualização** o próximo componente responsável pela concretização dos objetivos do projeto, permitindo a análise dos resultados armazenados.

3.3.3 Motor de visualização

O armazenamento da informação recolhida de forma periódica, não é de todo suficiente para tornar o processo de trabalho dos analistas mais eficiente. Embora nesta fase seja possível a consulta de todos os dados numa única plataforma, o processo de gestão e análise dos dados ainda é de longe o mais apelativo. É necessário criar meios de visualização de resultados práticos, não exaustivos e que permitam melhorar substancialmente o trabalho útil realizado.

O **motor de visualização** (Figura 3.9) é a componente do projeto que permite a criação de relatórios operacionais, com a informação estruturada e organizada para que possa ser consultada de forma fácil, intuitiva e dinâmica. Este componente organiza a informação armazenada na base de dados para facilitar a sua consulta. Este é um fator crucial para determinar o sucesso do resultado final do projeto, pois soluções que requerem um grande esforço de aprendizagem e adaptação, despendem de tempo que pode ser essencial para a realização de outras tarefas.

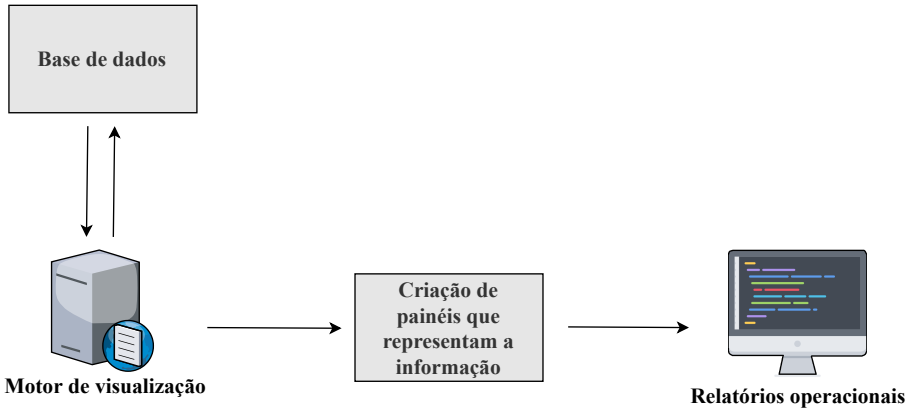


Figura 3.9: Representação do motor de visualização

Processo de geração de relatórios operacionais

Para a representação de resultados são criados relatórios dinâmicos, atualizados em tempo real de acordo com a informação previamente armazenada, tornando possível consultas com maior grau de especificidade. Deste modo, podem ser aplicados filtros que permitem consultar apenas determinados casos que podem ser relevantes à data da consulta, *i.e*, apenas ativos cujo risco é maior que determinado valor, ou ativos que apenas estão expostos à rede interna.

O formato dos relatórios foi definido com base na informação necessária à rápida interação com o problema, permitindo que os analistas consigam obter contexto, consequências e soluções de forma direta e esclarecedora. Para além da informação especifica de cada ativo ainda são apresentados dados que mostram o panorama geral da organização.

A informação disponível na consulta dos relatórios descreve:

- As características do ativo, isto é, o seu nível de exposição, a que grupo de ativos está associado (de acordo com o serviço que suporta), o risco e endereço IP.
- A distribuição dos graus de severidade das vulnerabilidades existentes.
- As vulnerabilidades detetadas, com a descrição das mesmas, o possível impacto, solução e categoria.
- Links de consulta a informação *open-source* acerca das vulnerabilidades.

3.4 Conclusão

Neste capítulo foi descrito de forma abstrata o CSVMS, incluindo os seus requisitos e arquitetura. Durante o processo de implementação, foram tomadas inúmeras decisões de acordo com os desafios encontrados, tentando sempre ir de encontro aos requisitos do projeto que serviram como linhas de guia para a obtenção de um produto final eficiente.

Inicialmente foram analisadas todas as possíveis componentes a incluir no projeto, bem como os ambientes em que iria ser realizado, para compreender de que forma cada uma poderia encaixar no produto final. Essa experiência motivou a criação de uma estrutura base que através da experimentação acabou por sofrer adaptações, resultando no modelo final acima descrito.

A ideia do cálculo do risco surgiu durante o processo de desenvolvimento do projeto, uma métrica que poderia vir a facilitar bastante o trabalho dos analistas da organização, e sobretudo algo que ainda não existia. Esse cálculo foi incluído no modelo do projeto para que se consiga compreender não só o seu propósito mas também como é constituído.

No próximo capítulo serão descritos em maior detalhe a implementação das componentes presentes no modelo do projeto.

Capítulo 4

Implementação

Neste capítulo é apresentada a descrição detalhada da implementação do projeto CSVMS, apresentando os problemas que surgiram, as decisões tomadas e as soluções técnicas adotadas para a sua concretização. A implementação deste projeto foi maioritariamente desenvolvida utilizando a linguagem de programação Ruby, ficando em sintonia com o modo de trabalho adotado pelas equipas de *devOps* da MEO. Também foram utilizadas ferramentas pertencentes ao *ELK Stack*[1] para auxiliar o armazenamento e representação de resultados, e a ferramenta *Blue Prism* [27] para desenvolver um processo de RPA que auxiliou na recolha de informação. Para além das ferramentas, grande parte das máquinas utilizadas correm sistemas operativos baseados em Linux e têm atribuídos privilégios de administração.

O CSVMS é constituído por três componentes principais: o **motor de calendarização**, o **motor de processamento** e o **motor de visualização**. Cada uma destas componentes é fulcral para o projeto e todas trabalham de forma coordenada para proporcionar uma solução operacional e robusta.

4.1 Motor de calendarização

O motor de calendarização (Figura 4.1) é a base de todo o projeto, pois assegura a execução das tarefas automatizadas a partir de um escalonamento definido. Uma vantagem desta componente é que opera automaticamente, não requerendo qualquer intervenção humana. Este módulo foi desenvolvido utilizando *cron jobs*, uma ferramenta que facilita o agendamento da execução de comandos, apenas necessitando que sejam configuradas as frequências de execução.

Considerando que existem várias fontes das quais serão extraídos resultados, idealmente a periodicidade de extração deveria ser baseada nas datas em que é atualizada a informação relacionada com as novas vulnerabilidades detetadas. No entanto, nem todos os motores de deteção e análise têm datas definidas, impossibilitando a adoção esta abordagem.

Para remediar esta situação foi definido que todas as segundas-feiras de madrugada (de forma intervalada para evitar problemas de escrita concorrente) a informação será extraída dos vários motores de deteção. A escolha deste dia vai de encontro com a publicação de resultados pela plataforma *Qualys*. Permite-se assim calendarizar as análises, garantindo que no início de cada semana é disponibilizada a informação atualizada da maior parte dos ativos. Caso algum dos restantes motores de deteção e análise de vulnerabilidades publique resultados no dia seguinte, no máximo terá de se aguardar uma semana para que se consiga analisar esta informação.

[1] Conjunto de ferramentas constituído pelo *Elasticsearch*, *Logstash* e *Kibana*

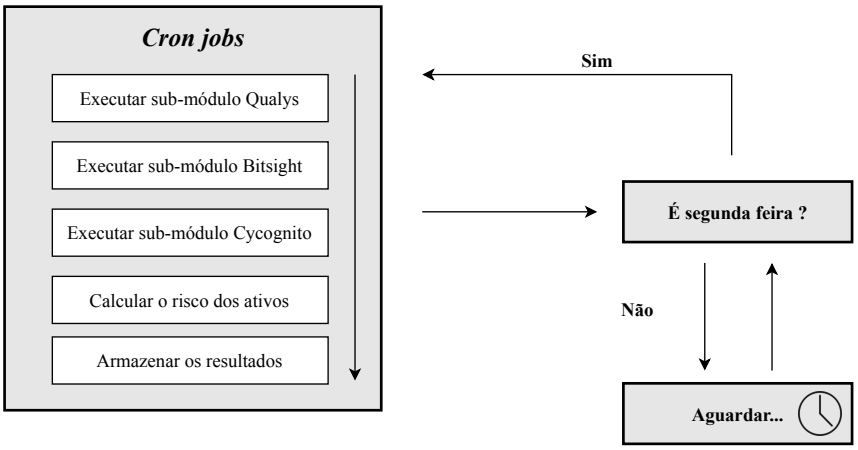


Figura 4.1: Representação do motor de calendarização

4.2 Motor de processamento

O motor de processamento é a componente responsável pela execução das interações com os MDAV's e pela concretização do processo de ETL. O seu desenvolvimento foi realizado de forma modular para garantir que cada interação com cada MDAV é realizada de forma independente, facilitando a correção de erros e a rápida alteração de qualquer operação. Com a abordagem adotada, o processo de adição de novas fontes de informação tornou-se mais simples, pois para incorporar uma nova fonte é somente necessário criar um novo módulo e o ajustamento de configurações.

A Figura 4.2 ilustra as várias etapas do processo realizado pelo motor de processamento, referenciando através de números os respetivos passos percorridos até que os objetivos desta componente estejam concluídos. É de notar que todas as caixas assinaladas a cinzento representam os constituintes do motor de processamento. Ao longo do capítulo serão apresentados os detalhes de implementação de cada um destes constituintes, explicando os meios técnicos utilizados para a sua concretização.

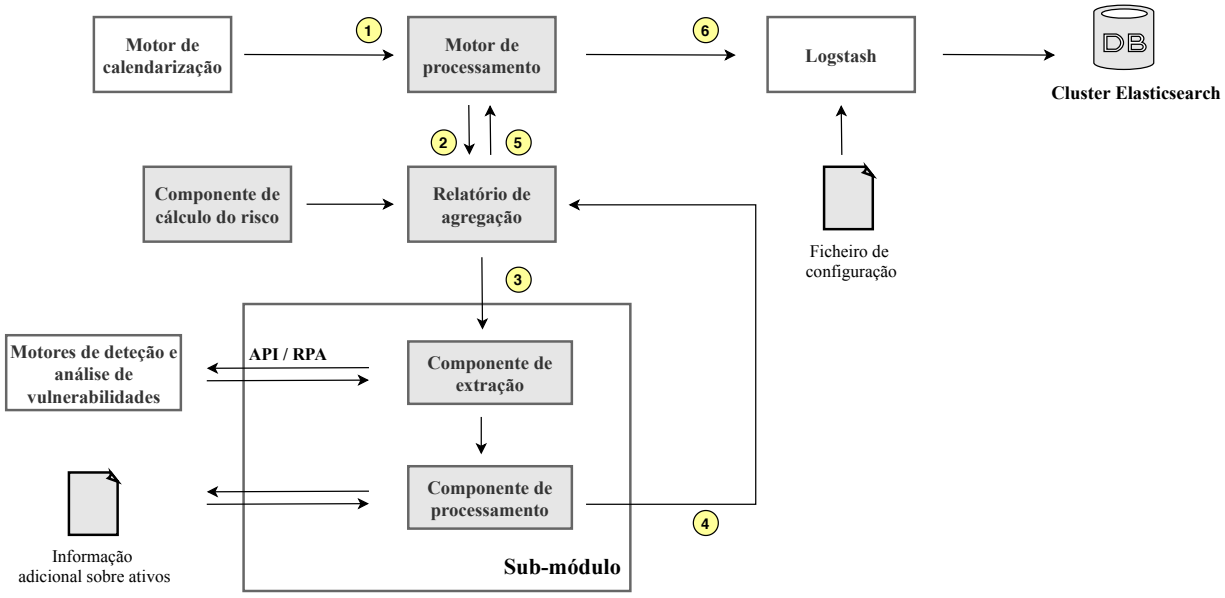


Figura 4.2: Etapas do processo realizado pelo motor de processamento

Assim sendo, o procedimento de trabalho é o seguinte:

1. É aguardada ordem do **motor de calendarização** para começo de extração de resultados.

2. Recebe a ordem de execução e é direccionado o pedido de acordo com o MDAV a interagir. No caso de ser a primeira extração da semana, é inicializado o relatório de agregação. Caso contrário, apenas é verificado se o relatório já existe e aguarda-se pelo sub-módulo respetivo.

3. É executado o sub-módulo correspondente ao MDAV que foi ordenada a extração. Este começa por recolher os dados, utilizando a componente de extração, que faz um pedido ao MDAV e armazena localmente a informação obtida. De seguida a componente de processamento é responsável por verificar os resultados armazenados e realiza a normalização e o enriquecimento dos mesmos.

4. São adicionados os dados processados ao relatório de agregação.

5. É indicado ao **motor de processamento** que o relatório já integra a nova informação. Se o sub-módulo que comunica for correspondente ao último MDAV a realizar a extração, este calcula imediatamente o risco de todos os ativos e armazena o valor juntamente com a informação. Caso contrário aguarda-se até o **motor de calendarização** ordenar a próxima extração.

6. Neste passo o relatório já deve ter centralizada toda a informação recolhida, sendo de seguida executada a ferramenta Logstash [25] para consumir o ficheiro e carregar os dados na base de dados implementada num *cluster* Elasticsearch [24].

4.2.1 Relatório de agregação

A inclusão de diversas ferramentas que produzem dados com formatos distintos, implica a abstração do CSVMS em relação ao formato original da informação. A abstração começa com a utilização de um relatório único, estruturado como sendo uma tabela que organiza a informação. Esta tabela inicialmente contém apenas as categorias que permitem indexar univocamente a informação e segue um formato canónico compatível com a eventual inclusão de novas ferramentas. No fim do processo de cálculo do risco, o relatório deverá ser fornecido como *input* de informação a carregar na base de dados.

Na criação do formato canónico, como mencionado no Capítulo 3, foi adotada maioritariamente a nomenclatura do MDAV considerado mais completo: o *Qualys*. Este abrange cerca de 60% dos resultados processados na MEO. As categorias adotadas servem essencialmente para dividir a informação de acordo com a sua temática, sendo que mais tarde ao ser carregado o ficheiro na base de dados, estas serão associadas à informação presente no campo respetivo.

O relatório é inicializado antes da primeira extração de informação semanal e serve como base para agregar toda a informação processada, sendo automaticamente eliminado após o armazenamento do respetivo conteúdo na base de dados. O tipo de ficheiro escolhido para a sua implementação foi o CSV, pois é um formato suportado pelo Logstash, para o qual existem disponíveis bibliotecas Ruby que facilitam quaisquer operações de manipulação estrutural necessárias de implementar. A Figura 4.3 ilustra um exemplo do seu funcionamento, em que está agregada informação proveniente de diferentes origens, indexada segundo uma única nomenclatura. Como é possível verificar esta tem presente na

primeira linha as categorias responsáveis pela indexação da informação e cada linha seguinte representa uma vulnerabilidade detetada para um determinado ativo.

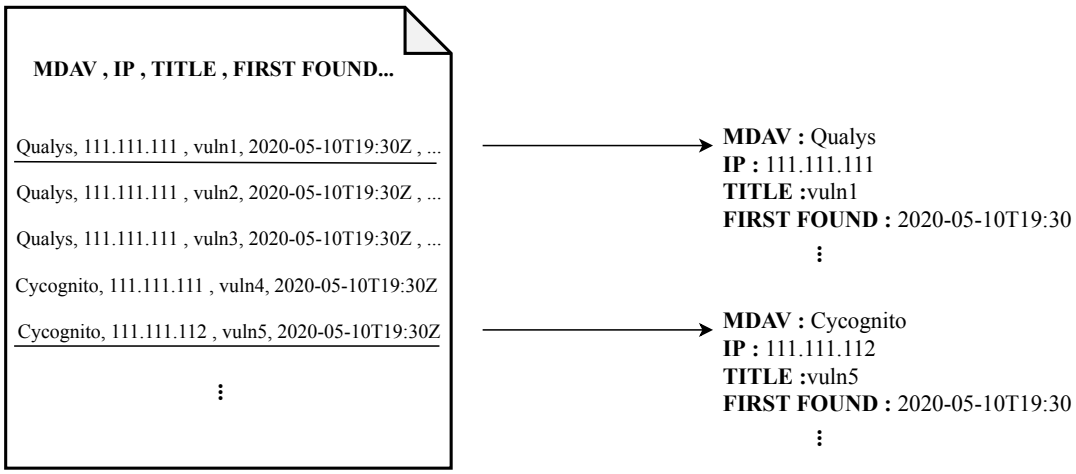


Figura 4.3: Exemplo de funcionamento do relatório de agregação

4.2.2 Sub-módulo - Implementação dos processos de extração do *Qualys* e *Bitsight*

A implementação dos processos de extração do *Qualys* e do Bitsight foi semelhante, pois ambos dispõem de API's que permitem a comunicação direta com o motor de processamento. A Figura 4.4 ilustra uma representação do procedimento de extração de informação único para cada MDAV. No caso do *Qualys*, a informação é transmitida no formato JSON e processada de seguida (não necessitando de armazenar localmente qualquer ficheiro). No caso do *Bitsight*, após o pedido ser concretizado, é transferido um ficheiro comprimido com a informação. Este ficheiro é imediatamente descomprimido para que se consigam processar os dados, e acaba por ser apagado após a agregação da informação no relatório único.

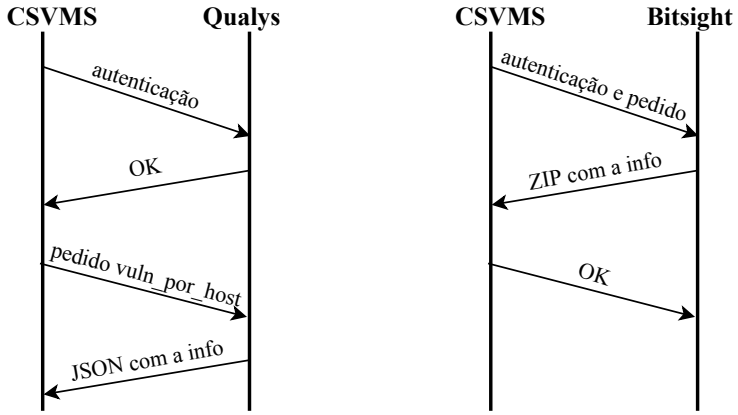


Figura 4.4: Casos de uso da extração de informação do Qualys e *Bitsight*

4.2.3 Sub-módulo - Implementação do processo de extração *Cycognito*

Ao contrário das outras ferramentas, no caso do *Cycognito* não está disponível qualquer API que permita estabelecer comunicações para extrair informação. O único método de extração disponibilizado é a interação manual com o *website* do MDAV, que permite descarregar um ficheiro CSV comprimido que contêm internamente os resultados armazenados. Esta limitação levou à procura de novas abordagens para realizar o processo de extração de informação, afim de tornar o projeto independente da interação humana.

A solução adotada no CSVMS passa pela implementação de um robô, através da tecnologia RPA, tendo este sido desenvolvido utilizando a ferramenta *Blue Prism*. O robô está representado na Figura 4.5 e funciona através de instruções definidas pelo utilizador, replicando interações com as aplicações abertas na máquina onde é executado. As interações imitam o procedimento que seria necessário para a realização da ação pretendida, e são definidas indicando passo a passo qual seria o processo que um humano teria de executar para conseguir atingir o seu objetivo na aplicação.

Neste caso apenas é necessária a interação com duas ferramentas, o navegador *Web* para aceder à página *Cycognito* e uma ferramenta que permita a transferência de ficheiros através de um protocolo seguro. Esta transferência é necessária, pois o processamento dos dados é realizado numa máquina de desenvolvimento que corre em Linux, enquanto os robôs têm de ser executados em máquinas Windows. Na Figura 4.5 estão representadas as tarefas a executar através de retângulos devidamente identificados, onde cada retângulo contém internamente as instruções para concretizar os objetivos pretendidos. A ordem de trabalhos no processo de extração do *Cycognito* é a seguinte:

1. É obtida informação acerca da máquina onde o robô é executado. Esta ação é necessária pois existem várias máquinas responsáveis pela execução de processos RPA, requerendo a abstração em relação a ambientes de trabalho concretos.

2. É aberto o navegador na página correspondente ao *Cycognito* e verifica se existe alguma sessão que ainda esteja aberta. No caso positivo, avança-se para o próximo passo, caso contrário é realizado o *login* através de credenciais de acesso armazenadas localmente de forma segura (cifradas utilizando AES-256).

3. Com sessão iniciada, é gerado o relatório com a informação do MDAV e de seguida este é descarregado para a máquina local.

4. É realizada uma verificação que aguarda até ao relatório estar presente na máquina local. Quando for detetado o ficheiro, fecha-se o navegador, altera-se o nome para que esteja de acordo com a verificação efetuada no sub-módulo e executa-se um *script* que envia o ficheiro para a máquina em que será processado.

É de notar que esta extração decorre automaticamente através da funcionalidade de calendarização própria da ferramenta *Blue Prism*. Para tal, foi agendada a execução semanal do robô, algum tempo antes da hora definida no **motor de calendarização** para o processamento do respetivo relatório. Assim é possível garantir que quando o sub-módulo do *Cycognito* é executado, este contem disponíveis os ficheiros necessários na máquina local. O sub-módulo é responsável por descomprimir o ficheiro extraído antes de processar a informação.

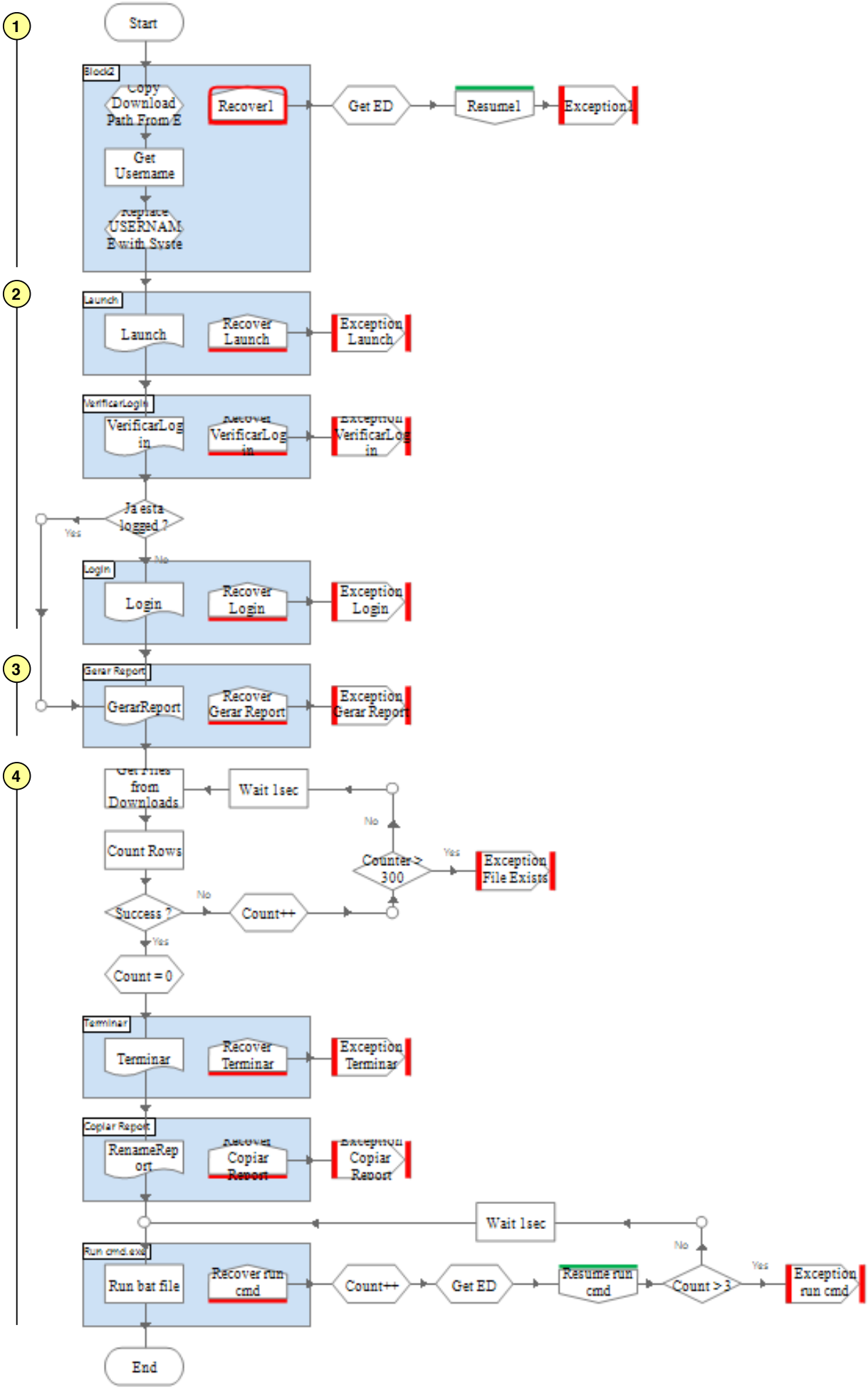


Figura 4.5: Caso de uso da extração de informação do Cycognito

4.2.4 Sub-módulo - Componente de processamento

Como referido no Capítulo 3, a componente de processamento de cada sub-módulo é responsável pelo processamento da informação obtida do MDAV correspondente. Esta componente itera sobre os dados recolhidos e para cada campo de informação, caso seja necessário, são convertidos os dados segundo um formato único. Deste modo obtém-se a homogeneização dos resultados que serão adicionados ao relatório de agregação.

Na primeira etapa de processamento é efetuada a leitura (linha a linha) da informação presente no relatório original, em que cada linha equivale a uma vulnerabilidade detetada sobre um dos ativos analisados. Para cada linha começa-se por identificar os atributos que separam a informação e é feita uma associação com a categoria equivalente existente no relatório de agregação, afim de identificar o tema que se está a tratar. Enquanto este processamento é executado, é feita uma cópia da informação que é armazenada numa estrutura de dados para que seja normalizada e depois escrita no relatório de agregação único. Esta estrutura de dados tem presente na primeira posição qual o motor de deteção e análise de vulnerabilidades utilizado, para que se consiga identificar facilmente a origem, como representado na Figura 4.6.

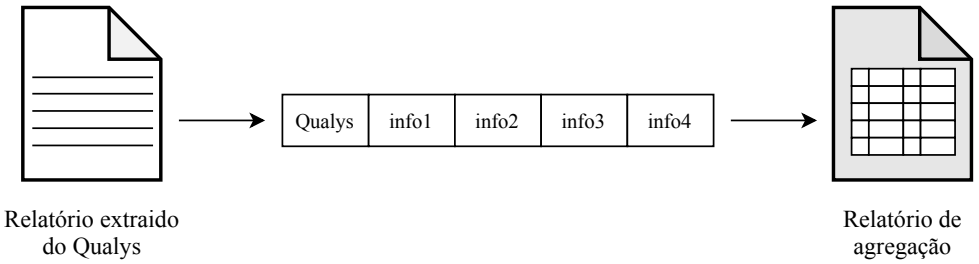


Figura 4.6: Exemplo do processamento da informação de um MDAV

Depois de identificada a informação é validado o tipo de dados utilizados para a representação dos valores. Estes são processados em dois casos, caso sejam **datas** ou caso seja o indicador da **severidade** da vulnerabilidade. No caso das datas o processo é simples dado que apenas é aplicada formatação para que seja apresentada de acordo com o formato da ISO 8061[28], ficando em conformidade com a norma internacional para representação de data e hora. Caso seja o indicador de severidade este é armazenado com o seu valor original, no entanto é necessário guardar também um valor normalizado para que seja calculado o risco mais tarde. Por fim são adicionados dois campos extra, o STORAGE DATE e o CSVMS ID . No primeiro guarda-se a data de extração dos resultados para que sirva de referência à semana em foi feito o processamento e no segundo guarda-se um valor único que identifica univocamente aquela vulnerabilidade (de forma iterativa).

Para além da normalização ainda é feito o enriquecimento da informação conhecida sobre os ativos, adicionando ainda mais informação sobre os serviços que estes suportam. Este processo é realizado através da consulta de um ficheiro auxiliar que contém informação agregada sobre os endereços IP dos ativos da MEO, descrevendo que tipo de serviços são executados e a sua importância para a empresa.

Esta informação adicional é também fulcral para o processo de cálculo do risco. Sendo estruturada no ficheiro auxiliar de acordo com a seguinte estrutura:

- **Endereço IP -** Índice que descreve o endereço IP associado ao ativo;

- **Grau de importância para a empresa -** Contém um valor que representa a importância do serviço prestado pelo ativo para a empresa. Este valor segue uma escala própria a que foram atribuídos pesos consoante o grau de relevância;
- **Serviço Associado -** Descreve por extenso o tipo de serviço associado ao ativo, *i.e*, *DNS Resolver*;
- **Grau de exposição -** Índice que descreve se o ativo está ou não exposto à rede externa da organização. Também contabilizado na fórmula do risco.

Assim, para cada linha processada é feita a pesquisa do IP no ficheiro de enriquecimento e é recolhida a informação que também será depositada na estrutura de dados, para que no fim seja armazenada juntamente com toda a informação no relatório de agregação.

Normalização dos valores de severidade

A severidade das vulnerabilidades encontradas é uma das métricas atribuídas pelos MDAV's e tem o objetivo de classificar a criticidade da existência de cada vulnerabilidade no ativo em que foi detetada. Esta métrica pode ser representada seguindo diferentes escalas, tanto quantitativas como qualitativas e permite não só perceber quais as fragilidades mais criticas mas também definir uma ordem de prioridade na sua resolução. Os elementos que são tidos em conta para a atribuição deste valor variam de acordo com o MDAV. No entanto, é comum considerar fatores como a facilidade de exploração, existência de informação publicamente disponível para a sua exploração e o possível impacto.

Para conseguir formular um índice de priorização na resolução de vulnerabilidades, neste caso a métrica **risco**, foi necessário considerar a severidade de todas as vulnerabilidades detetadas por cada ativo, obrigando à normalização dos valores numa só escala. Esta escala está representada na Figura 4.7 e é transversal ao MDAV que originou o valor.

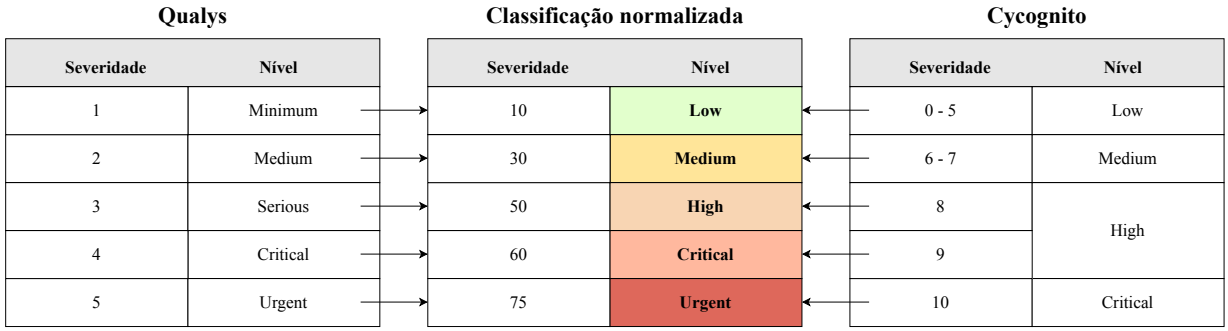


Figura 4.7: Modelo de classificação normalizado.

Das ferramentas utilizadas apenas foram considerados o *Qualys* e o *Cycognito* como fundamentais para a atribuição do nível de severidade. Já a *Bitsight* não foi incluída na normalização devido ao seu método de operação mais informativo, que apenas faz leituras das configurações e métricas dos endereços alvo, comparando-os com a média da indústria.

Dado que o valor normalizado apenas serve para auxiliar o cálculo do **risco**, este é armazenado juntamente com a severidade original. Assim torna-se mais fácil aos analistas compreender o tipo de vulnerabilidades que estão a ser apresentadas.

4.2.5 Cálculo do risco dos ativos

O cálculo do risco é efetuado antes de serem armazenados os dados contidos pelo relatório de agregação na base de dados. Este processo começa por fazer uma leitura integral do relatório e guarda em memória todos os valores necessários à fórmula de cálculo, associando-os ao respetivo endereço IP. Para cada endereço IP são armazenados o número de vulnerabilidades existentes (número de linhas com o mesmo endereço IP), a média das severidades das vulnerabilidades, o grau de importância do ativo para a empresa e também a exposição do mesmo para a rede exterior.

Com todas estas variáveis presentes é multiplicada cada uma delas por um peso proporcional ao risco de exploração por agentes maliciosos. O objetivo desta multiplicação é conseguir distinguir de forma clara o risco associado a ativos com diferentes características, obtendo no fim um valor numérico robusto. Após a aplicação dos pesos é formulada a equação de cálculo e armazenam-se os valores (risco bruto e categoria de risco) no relatório de agregação juntamente com o resto da informação.

4.2.6 Implementação do armazenamento de informação na base de dados

O processo de armazenamento da informação é realizado através da ferramenta Logstash. Esta ferramenta é responsável por consumir a informação que se pretende armazenar no *cluster Elasticsearch* e requer o desenvolvimento de um ficheiro de configuração que especifica as características do caso de uso a realizar. O ficheiro de configuração é estruturado de acordo com o seguinte formato:

- ***input*** - Neste campo é especificada a origem dos dados a consumir e o modo de leitura a aplicar, *i.e*, o tipo de ficheiro em que está armazenada a informação e o *path* respetivo. De acordo com o formato também é especificado o modo de leitura, indicando padrões de separação de informação e número máximo de linhas consecutivas;

- ***filter*** - Neste campo é efetuada estruturação dos dados, organizando-os de acordo com as suas características. Primeiro é indicado qual o índice da base de dados a que será associada a informação armazenada. Já com o índice definido indicam-se os parâmetros que vão organizar a informação: as categorias presentes no ficheiro carregado que separam a informação, as mutações a realizar e os formatos dos diferentes tipos de dados de acordo com a sua categoria, *i.e*, quais as categorias que representam datas, números ou texto;

- ***output*** - Neste campo é especificado o destino da informação processada, tipicamente o endereço do *cluster Elasticsearch* onde serão armazenados os dados, o porto respetivo e o *path* local dos certificados necessários.

Este processo ocorre após o cálculo do risco dos ativos ao executar a ferramenta *Logstash* sobre o relatório de agregação, para que sejam carregados todos os seus dados de uma só vez. Quando os dados dão entrada no *cluster Elasticsearch* têm de ser associados a um índice, um mecanismo de organização de dados. Os índices são comuns em bases de dados não relacionais e permitem separar a informação de acordo com a fonte de origem e propósito. No caso do projeto CSVMS, o índice definido para o *cluster* Elasticsearch foi **qa.vam-csvms.events**.

4.2.7 Implementação do *software*

A solução concebida para concretizar os objetivos do motor de processamento foi desenvolvida de raiz, ponderando a abordagem a seguir de acordo com os requisitos de projeto e tendo em conta o ambiente operacional existente na MEO. A Figura 4.8 ilustra o diagrama de classes UML que facilita a compreensão da estrutura do *software* implementado.

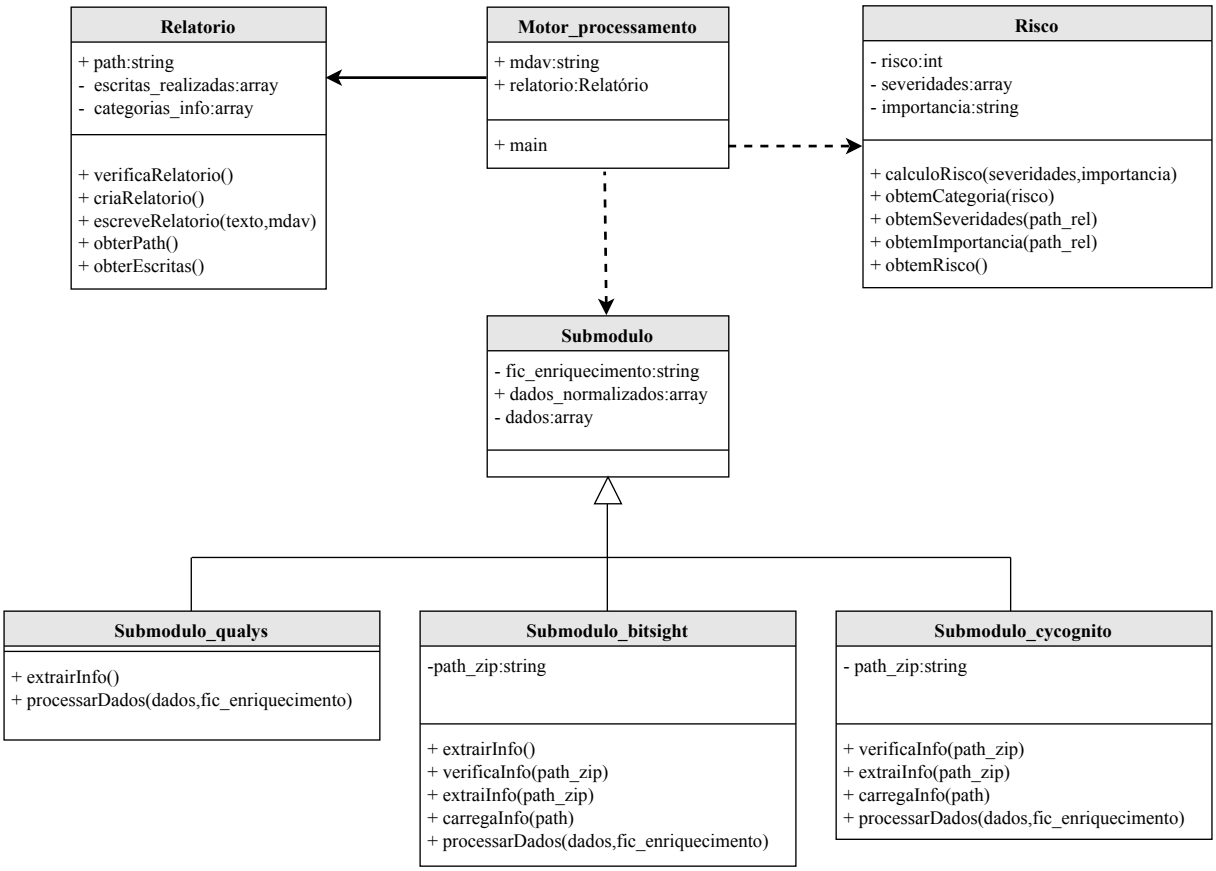


Figura 4.8: Diagrama de classes do motor de processamento

Classes incluídas no diagrama

- **Motor_processamento** - Recebe como *input* o MDAV a executar e é responsável por orquestrar as atividades necessárias para que a informação seja extraída, processada e agregada no relatório. No caso de não receber qualquer *input*, significa que todos os sub-módulos já foram executados anteriormente, procedendo ao cálculo do risco para depois carregar resultados na base de dados.

- **Relatorio** - Define o conceito de relatório de agregação para o projeto, permitindo a manipulação direta do ficheiro que persiste a informação. São guardadas localmente as categorias definidas para que sejam escritas durante a criação do relatório.

- **Risco** - Classe responsável pelo cálculo do risco. Disponibiliza operações de consulta ao ficheiro do relatório para obtenção de variáveis necessárias. Calcula o risco e devolve-o ao motor de processamento para que o armazene juntamente com o resto da informação.

- **Submodulo** - Super-classe que define a estrutura comum a cada sub-módulo. As classes derivadas definem caso a caso as operações necessárias para o correto funcionamento segundo o motor de deteção e análise de vulnerabilidades a que estão associadas. Contêm as operações necessárias para extração (exceto no *Cycognito*) e para processamento e enriquecimento dos dados.

De acordo com a ordem de execuções previamente descrita, estas classes definem a implementação da componente do motor de processamento. É de notar que todas as verificações e tratamentos de erros são realizadas quando necessário pelo que não foi considerado relevante descrever em detalhe no contexto do projeto.

4.3 Motor de visualização

O motor de visualização é a terceira componente do projeto e é responsável pela criação de métodos de visualização de resultados, produzindo relatórios operacionais que permitam aos analistas de ciber segurança fazer a consulta quando necessário. Esta componente foi desenvolvida com base na ferramenta *Kibana* [26], permitindo interagir e criar representações visuais através dos dados que foram carregados no *cluster Elasticsearch*.

Sobre os dados armazenados no índice **qa.vam-csvms.events** (Figura 4.9), foram criadas várias *visualizations* que são agregadas num único *dashboard*, criando assim o **relatório operacional** (ver Anexo A.1). Uma *visualization* é uma representação visual dos dados armazenados nos índices *Elasticsearch* e pode ser criada utilizando diferentes estruturas visuais (histogramas, tabelas, gráficos). O propósito das *visualizations* é guardar automaticamente as *queries* que procuram os dados a pretendem representar. Um *dashboard* é uma coleção de *visualizations* que permite a consulta das mesmas numa só página. Para além da consulta ainda permite a interação direta com os resultados, possibilitando a aplicação de filtros dinâmicos que alteram os dados apresentados de acordo com a pesquisa que se está a realizar.

Figura 4.9: Resultados presentes no índice do projeto CSVMS

O relatório operacional deve permitir a consulta de informação técnica mas também possibilitar a compreensão do panorama geral das vulnerabilidades na organização. A ideia é criar um método de consulta simples e direto, que permite rapidamente visualizar a informação mais importante. Para tal, foram desenvolvidas algumas *visualizations*, que permitem representar os dados de diversas formas. As *visualizations* podem tanto ser de carácter estatístico (Figura 4.10) como tabelas que representam texto em bruto. O objetivo é complementar a informação a apresentar, oferecendo uma solução visualmente agradável.

No entanto, caso seja necessária uma consulta mais específica que a disponibilizada pelo relatório, o analista pode pesquisar diretamente no *Kibana* e aceder à informação em bruto, relativa à vulnerabilidade em questão, como representado na Figura 4.11. Como é possível verificar estão descritos os vários campos recolhidos da ferramenta original, bem como o **risco** do ativo (*asset_risk*) e a **categoria atribuída ao risco** (*asset_risk_class*).

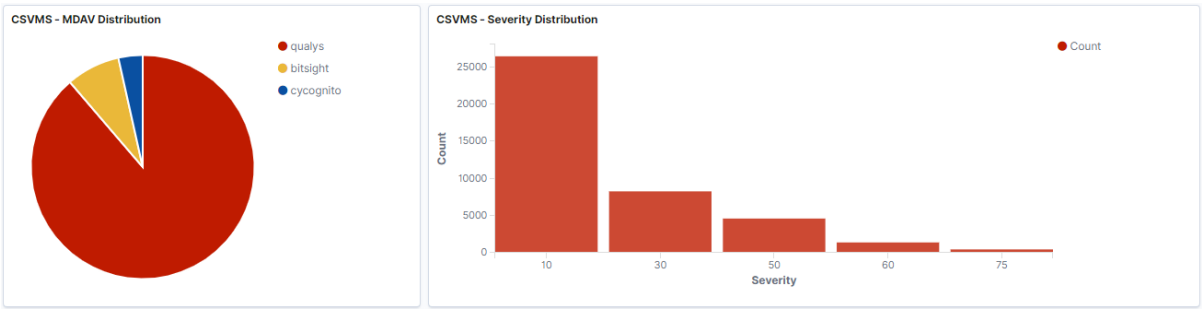


Figura 4.10: *Visualizations* criadas para representar dados

t asset_groups	HIDRA MGMT Asset Group
# asset_risk	43.85
# asset_risk_class	3
t bugtraq_id	100283 URL:http://www.securityfocus.com/bid/100283
t category	RedHat
# csvms_id	2163
t cve_id	CVE-2017-1000117 URL:http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000117
# cvss3_final	7.9
# cvss_final	5.3
t dns	[redacted]
first_found	05-09-2017 11:15:58 +0100
t impact	An attacker could use this flaw to execute shell commands with the privileges of the user running the Git client, for example, when performing a action on a malicious repository or a legitimate repository containing a malicious commit.
t ip	[redacted]
last_found	05-06-2018 11:14:28 +0100
t mdav	qualys
t mdav_severity	Critical
t operating_system	Red Hat Enterprise Linux Server 6.8
t platform_issue_id	QID236462
t result	tablePackage Installed Version Required Version git 1.7.1-4.el6_7.1.x86_64 1.7.1-9.el6_9 perl-Git 1.7.1-4.el6_7.1.noarch 1.7.1-9.el6_9
# severity	60
t solution	Upgrade to the latest packages which contain a patch. Refer to <A HREF=https://access.redhat.com/articles/11258 TARGET=_blank>Applying Package Updates to RHEL system</A> for details. <P> Refer to Red Hat security advisory <A HREF=https://access.redhat.com/errata/RHSA-2017:2485 TARGET=_blank>RHSA-2017:2485</A> to address this issue and obtain more information. <P>Patch: Following are links for downloading patches to fix the vulnerabilities:

Figura 4.11: Representação integral da informação de uma única vulnerabilidade no *Kibana*

4.4 Conclusão

Neste capítulo foram descritos todos os detalhes considerados para a implementação do CSVMS no ambiente operacional da MEO. Estes detalhes foram essenciais para assegurar que o projeto cumpre os requisitos definidos inicialmente, e também para possibilitar a utilização do sistema no dia a dia da empresa. Foram ainda apresentados alguns constituintes do relatório que agrega a informação, e que servirá como meio de consulta pelos analistas de segurança.

No próximo capitulo serão realizados testes para avaliar se os requisitos foram cumpridos e para perceber se o resultado final é satisfatório.

Capítulo 5

Resultados e Avaliação

Neste capítulo é realizada a avaliação e análise do sistema desenvolvido para o projeto CSVMS. O propósito desta análise é perceber se os objetivos estabelecidos inicialmente foram alcançados e se os requisitos operacionais (Secção 3.1) foram cumpridos. Para tal, foram utilizadas duas abordagens distintas: foi realizado um **questionário** sobre a usabilidade da solução e foi também **implementado um caso de uso** sobre um dos MDAV's não incluídos no projeto.

5.1 Questionário de usabilidade

Para entender a reação dos utilizadores ao utilizarem o CSVMS, foi utilizado um questionário (Anexo B.1), que segue o modelo de classificação de usabilidade *System Usability Scale* (SUS) [29]. Este modelo procura classificar e quantificar a usabilidade das soluções analisadas através da aplicação de uma metodologia, que permite obter resultados transversais ao tipo de sistemas que é avaliado. Para tal, são consideradas métricas subjetivas ao tema do projeto que permitem compreender a **eficácia**, a **eficiência** e a **satisfação** na sua utilização.

O questionário é composto por dez afirmações diretas que são respondidas de acordo com a opinião do questionado após utilizar a solução CSVMS. Cada afirmação tem associada uma escala entre 1 e 5 que permite expressar concordância ou discordância em relação à mesma. O objetivo será não só entender a opinião dos questionados mas também encontrar pontos de discordância entre as opiniões, para que sejam estudados e compreendidos. Após a recolha de resultados será também calculada uma pontuação SUS que quantifica a usabilidade da solução que está a ser estudada. Este valor é gerado através da fórmula própria do modelo, que considera todas as respostas. No caso das perguntas pares, é subtraído cada valor a 5, no caso das perguntas ímpares é subtraído 1 à resposta do utilizador. Por fim, é realizado o somatório de todas as respostas, normalizadas segundo o passo anterior, e é multiplicado o valor por 2,5 obtendo a pontuação SUS que estará compreendida entre 0 e 100.

5.1.1 Participantes

Dado que o CSVMS foi desenvolvido seguindo os costumes operacionais da MEO (métodos de trabalho, ferramentas utilizadas e processamento de informação confidencial), não foi possível estender a participação no preenchimento do questionário a uma grande amostra populacional. Portanto, foram apenas considerados técnicos especialistas na área de segurança informática, pertencentes à equipa de *Vulnerability Assessment and Management* (VAM) da MEO. Esta equipa é responsável pela gestão de

vulnerabilidades de toda a infraestrutura da empresa, está altamente familiarizada com a utilização dos MDAV's incluídos no projeto e tem como função principal tornar a MEO tão segura quanto possível.

Os questionários foram aprovados internamente pela MEO e de seguida foram partilhados *online*. Foram recolhidas respostas de 10 participantes que integram a equipa acima descrita. O preenchimento do questionário tem uma duração de aproximadamente 5 minutos e foi realizado sem existir qualquer discussão prévia acerca do método de funcionamento do projeto, sendo apenas fornecido o relatório gerado pelo CSVMS para consulta (Anexo A.1). O objetivo pretendido foi perceber qual seria a reação da equipa responsável pelo tratamento de vulnerabilidades na MEO, ao substituir o atual procedimento de análise pela consulta dos resultados produzidos pela solução desenvolvida.

5.1.2 Resultados

A Tabela 5.1 representa as respostas recolhidas de todos os participantes. Deste modo foi elaborado um *heatmap* que organiza as respostas dos participantes face às perguntas do questionário. Este *heatmap* permite visualizar facilmente o índice de concordância dos participantes bem como o de neutralidade, simplificando a análise das respostas.

Questões	Part. 1	Part. 2	Part. 3	Part. 4	Part. 5	Part. 6	Part. 7	Part. 8	Part. 9	Part. 10
1. Imagino-me a utilizar esta solução com frequência.	4	5	4	3	4	1	3	3	5	5
2. Achei a solução desnecessariamente complexa.	1	1	2	2	1	1	1	3	1	1
3. Achei a solução fácil de utilizar.	4	4	5	5	5	5	5	4	5	4
4. Preciso de apoio de um técnico para recorrer a esta solução.	1	2	2	3	1	1	2	2	1	1
5. Achei que as diversas funções da solução estavam bem integradas.	3	5	4	4	5	5	5	4	5	4
6. Achei que existia demasiada inconsistência na solução.	2	1	1	1	2	2	1	2	1	1
7. A maioria das pessoas irá aprender a utilizar a solução rapidamente.	5	5	5	5	5	5	4	4	4	4
8. Achei a solução demasiado pesada para ser utilizada.	1	2	1	2	1	1	1	3	1	1
9. Senti-me bastante confiante ao utilizar a solução.	4	5	4	3	5	4	4	3	4	4
10. Tenho de aprender muitos conceitos antes de poder utilizar a solução.	1	3	3	3	3	3	2	3	2	1

Tabela 5.1: *Heatmap* de distribuição de respostas do Questionário SUS.

Como é possível verificar, a grande maioria das respostas foi positiva (assinalada a verde), o que demonstra que a solução foi no geral considerada bem definida e adequada. Contudo, existem alguns casos excecionais, onde se verificou a existência de alguma neutralidade/discordância de opiniões:

- No caso da 1ª questão, que avalia a frequência de utilização, existiu uma única resposta contraditória (assinalada a vermelho). Esta resposta deve-se ao facto do participante 6, ainda que pertencente à equipa de VAM, não ser responsável atualmente pelo processo de análise de vulnerabilidades. Como não tem essa responsabilidade, não considerou necessária a utilização do CSVMS no seu dia a dia;

- No caso da 10ª questão, que representa a quantidade de conceitos necessários a aprender para utilizar a solução, as respostas foram maioritariamente neutras. Este facto deve-se à utilização do *Kibana* para produção de relatórios operacionais, ferramenta que ainda é relativamente recente para a maioria dos técnicos.

A pontuação SUS média de todas as respostas é de **82.25**, demonstrando ser muito satisfatória. Este valor representa a pontuação classificativa da usabilidade da solução. Para além do *heatmap* anterior, ainda foram construídos dois gráficos que permitem ter a perceção da pontuação SUS individual, tanto por questão realizada como por participante (Figuras 5.1 e 5.2).

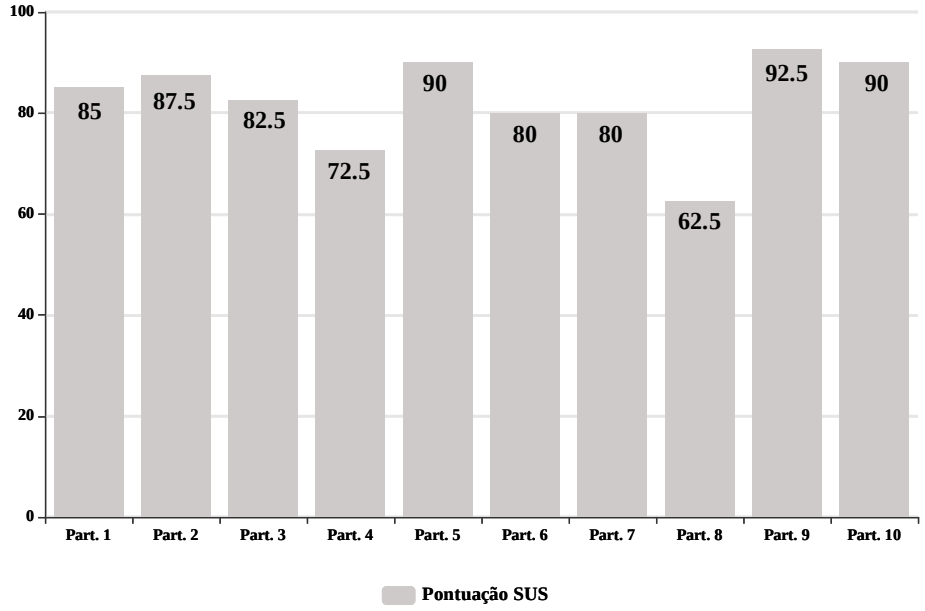


Figura 5.1: Distribuição da pontuação SUS por participante.

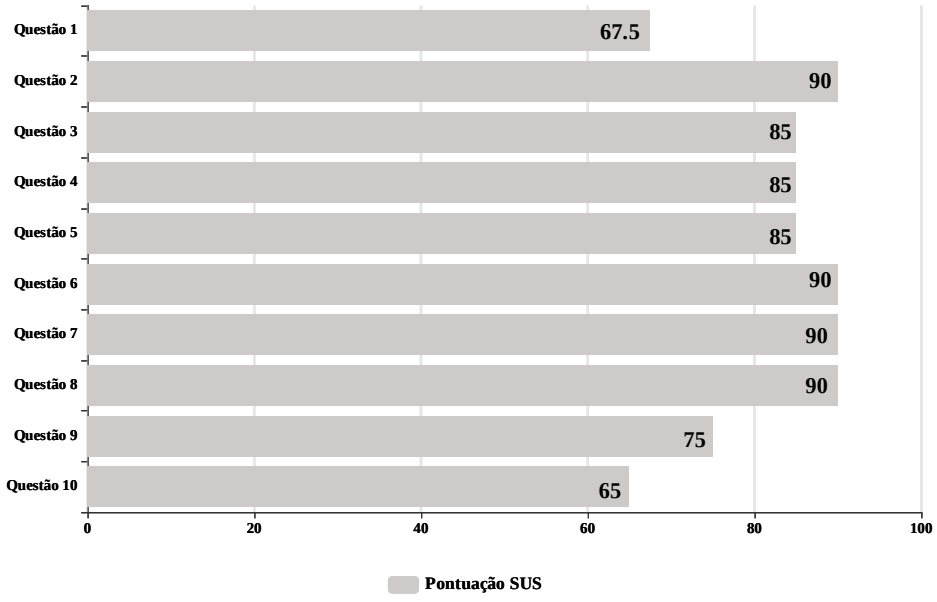


Figura 5.2: Distribuição da pontuação SUS por questão.

Ao analisar ambos os casos, particularmente onde se verifica menores pontuações, é possível constatar que a adoção de uma nova solução no meio operacional causa alguma resistência. Esta resistência deve-se ao facto de para além da adoção de uma nova tecnologia, anteriormente mencionada, qualquer novo sistema implementado no procedimento diário de uma organização tem uma curva de aprendizagem até que exista a completa familiarização. No caso específico do participante 8, é de notar que foi responsável pela pontuação SUS mais baixa (62.5), respondendo de forma neutra a metade das perguntas. Este caso ocorreu devido às tarefas pelas quais o participante é responsável, ainda que pertencente à equipa de gestão de vulnerabilidades, tem associadas outras funções que não a análise da informação técnica. Mesmo assim, os resultados do questionário permitem concluir que este processo de aprendizagem não terá um elevado grau de dificuldade e que existe elevado grau de aceitação da solução.

5.1.3 Discussão

O propósito da elaboração do questionário incide sobre a vontade de perceber se os requisitos relacionados com a experiência de utilização, foram respeitados. A análise aos resultados apresentados acima demonstra que a solução foi bem recebida e foi atribuída uma pontuação SUS bastante elevada.

Após a análise dos resultados acima obtidos é possível verificar o cumprimento de três dos requisitos inicialmente definidos (ver Secção 3.1):

- **Simplicidade e facilidade de utilizar** - Foi definido que para que o sistema fosse aplicável em contexto empresarial, este teria de ser fácil e intuitivo de utilizar. Este requisito deve-se ao objetivo principal do projeto que é tornar mais eficiente o processo de gestão de vulnerabilidades. Como é possível verificar ao analisar as respostas das questões 1-4,7,9 e 10, a solução foi considerada fácil e descomplexa, existindo bom *feedback* acerca da possibilidade de utilização no futuro;

- **Qualidade da informação** - Dada a necessidade de apresentar a informação tão objetiva e útil quanto possível, que permitirá aos analistas inspecionar e encaminhar a informação das vulnerabilidades de forma mais rápida. Através da análise das respostas às questões 1,5,6 e 9, podemos considerar positiva a implementação deste requisito;

- **Eficiência do sistema** - O acesso rápido à informação, sem grandes tempos de espera, é essencial para a correção mais rápida das vulnerabilidades. As respostas das questões 1,5,8 e 9, permitem concluir que o CSVMS é rápido e eficiente, não apresentando resultados que se associam à desmotivação derivada do tempo de espera necessário para que sejam produzidos resultados.

5.2 Implementação do novo caso de uso

A implementação de um novo caso de uso, surge como método de avaliação à adaptabilidade e escalabilidade do sistema. Inicialmente, como mencionado na Secção 3.2, foram considerados vários motores de deteção e análise de vulnerabilidades para incluir no CSVMS. No entanto, como se explicou anteriormente, algumas opções acabaram por ser excluídas da solução final.

Para conseguir implementar o novo caso de uso, foi necessário consultar novamente a lista inicial de MDAV's e escolher mais uma opção, que fosse útil incluir no CSVMS. A decisão final foi a implementação dos resultados produzidos pelo *Mozilla Observatory*. Esta escolha foi limitada pelas

opções que estavam em condições de ser utilizadas internamente e considerou também o motivo de exclusão inicial, que neste caso, seria apenas pela baixa frequência de utilização. O *Mozilla Observatory* tem como propósito analisar as configurações HTTP, através do domínio fornecido, e pode ser implementado para enriquecer a informação disponível no repositório. É de notar que este caso de uso não foi incluído como variável no cálculo do risco, pois a informação produzida é apenas de carácter informativo.

5.2.1 Implementação

O processo de implementação do novo caso de uso, correspondente ao *Mozilla Observatory*, foi divido em três fases: **investigação**, **desenvolvimento** do respetivo sub-módulo e **verificação** da integridade do sistema. Em cada fase foi medido o tempo despendido, para que seja possível compreender o quão fácil é a inclusão de novos motores de deteção e a análise de vulnerabilidades no CSVMS.

A fase de investigação baseia-se na obtenção todo o conhecimento necessário sobre o MDAV, para que se consiga desenvolver posteriormente o sub-módulo. Nesta fase o objeto principal é perceber qual o método de extração a utilizar, qual o *input* necessário ao funcionamento, e qual o formato do *output* produzido. Em relação ao método de extração, rapidamente foi encontrada documentação relativa à API disponibilizada pelo próprio MDAV. Esta API permite comunicar com o *Mozilla Observatory* e obter resultados de acordo com o domínio incluindo no pedido. Após a comunicação com o MDAV é necessário perceber qual a estrutura dos resultados obtidos para garantir que a informação é corretamente mapeada no relatório de agregação, excluindo informação redundante.

O sub-módulo foi desenvolvido de acordo com a informação recolhida previamente. Primeiro foi desenvolvido o método de extração tirando partido da API. Esta teve de ser *whitelisted* internamente para que as comunicações não fossem bloqueadas. Após a integração da mesma, foi definido que o sub-módulo teria de fazer um pedido por cada domínio de ativos considerados críticos. Estes pedidos têm de ser feitos de forma individual, processando depois a informação obtida para que enriqueça o relatório de agregação. O processamento dos resultados baseou-se sobretudo na separação da informação segundo as categorias previamente definidas, para facilitar o processo de consulta dos analistas.

Por fim, já com o sub-módulo desenvolvido, foram executados testes para verificar a integridade do sistema após a inclusão dos resultados do novo MDAV. O sistema foi executado novamente, e foram corrigidos todos os erros identificados.

5.2.2 Tempo decorrido

Como mencionado anteriormente, durante todo o processo de implementação foi medido o tempo necessário a realizar cada uma das fases. Os registos obtidos podem ser consultados na Tabela 5.2, representados de forma estruturada e objetiva.

A fase inicial de investigação foi pouco demorada devido à simplicidade da ferramenta. Para além disso existe muita informação *online* que ajuda a perceber qual a melhor abordagem para incluir os resultados do MDAV noutras soluções. A familiarização com a ferramenta incluiu toda a aprendizagem necessária de funcionamento, mas também a análise dos resultados obtidos. Ao analisar estes dados foi possível preparar de forma mais eficiente a execução da próxima fase.

O desenvolvimento do sub-módulo foi a fase da implementação mais curta de todas, pois foi possível seguir o modelo genérico utilizado em outros sub-módulos. Este modelo permite agilizar a componente

de processamento, que poderia demorar mais tempo visto que é necessário mapear a informação. Em relação à componente de extração, o facto de existir uma API bem definida e documentada, permitiu que o desenvolvimento do processo fosse rápido e direto.

Com o sub-módulo operacional, foram efetuados dois testes para garantir que não existiam quaisquer erros derivados da integração do novo caso de uso. Após o primeiro teste foram identificados dois erros que foram resolvidos de imediato. Com a inclusão da correção e na segunda execução do CSVMS já não foram encontrados quaisquer erros, verificando-se o sistema completamente funcional. É de notar que ainda que tenham sido gastas 1 hora e 3 minutos durante este processo, 46 minutos resultaram das execuções do próprio sistema (cada execução demora aproximadamente 23 minutos).

Fase	Tempo
Investigação acerca do MDAV	**1h 15m**
- Familiarização com a ferramenta	1h 02m
- Estudo da API	13m
Desenvolvimento do sub-módulo	**56m**
- Componente de extração	22m
- Componente de processamento	34m
Verificações e testes de integridade	**1h 03m**
- Execução do CSVMS	46m
- Correção de erros	17m
Tempo total≈	**3h 23m**

Tabela 5.2: Registo do tempo decorrido durante a implementação do novo caso de uso.

5.2.3 Discussão

O tempo total de trabalho necessário a incluir o novo motor de deteção e análise de vulnerabilidades foi aproximadamente 3 horas e 23 minutos. Este registo é bastante positivo na medida em que, após a implementação do caso de uso, todas as próximas extrações terão incluídos os resultados do MDAV. Estes resultados, ainda que sejam apenas necessários consultar em casos específicos, encontram-se agora disponíveis para consulta evitando desperdício de tempo na análise manual.

Através da análise dos tempos obtidos e com a confirmação de que é possível incluir um novo MDAV no sistema, podemos concluir que os seguintes requisitos foram cumpridos para o *Mozilla Observatory*:

- **Modularidade do sistema** - Foi definido que o sistema não deveria depender de qualquer um dos motores de deteção e análise de vulnerabilidades para funcionar. Este requisito é validado através da arquitetura definida para a solução, e pode ser confirmado através da implementação do novo caso de uso. Ao verificar que a inclusão de um novo MDAV apenas requer o desenvolvimento do sub-módulo correspondente, independente a todos os outros, produz evidência que o sistema é modular.

- **Escalabilidade do sistema** - Foi definido inicialmente que para garantir a continuidade do CSVMS, teria de ser possível adicionar novos MDAV's ao sistema. Ao incluir um novo caso de uso, já sobre a versão final do projeto, é possível verificar que o sistema permite integração de novas fontes de informação. Possibilitando também a adição de novos alvos a analisar.

- **Compatibilidade com novas ferramentas** - A inclusão de novas ferramentas tem de ser um processo fácil e rápido. Este processo, como descrito acima, tira partido de toda a infraestrutura existente, e necessita apenas da reutilização de um modelo geral de sub-módulo (o mesmo utilizado em todos os outros casos). Esta possibilidade de reutilizar o padrão desenvolvido permite tornar mais rápido o processo de adição de informação, e assim validar este requisito inicialmente definido. É de notar que foram apenas necessárias cerca de 3 horas para adicionar a nova ferramenta, o que é bastante positivo tendo em comparação com o tempo que teria de ser despendido em análise manual.

5.3 Conclusão

Apesar de não existir uma base de comparação empírica, baseada no desempenho de um processo manual idêntico de gestão de vulnerabilidades, este capítulo permitiu avaliar duas componentes fundamentais do CSVMS. A avaliação da experiência de utilização demonstrou que o sistema foi bem aceite pelos técnicos de cibersegurança. Para além da simplicidade de utilização, a informação produzida foi considerada útil e prática. A avaliação da componente técnica, ao incluir um novo MDAV, demonstrou que o sistema permite integrar rapidamente novas ferramentas.

Em suma, o CSVMS obteve resultados positivos e poderá ser considerado como futura abordagem para otimizar o processo de gestão de vulnerabilidades interno.

Capítulo 6

Conclusão e Trabalho Futuro

6.1 Conclusão

Esta tese apresenta um sistema de gestão de vulnerabilidades que visa otimizar o atual procedimento realizado pela MEO. Este sistema foi desenvolvido com base na análise das etapas do processo corrente, identificando os respetivos pontos fracos. Posteriormente, foram definidas melhorias para que fosse possível minimizar o desperdício de recursos em tarefas desnecessárias.

Como resultado, foi desenvolvido o CSVMS, um sistema que funciona como repositório de vulnerabilidades dinâmico e que permite a consulta de informação transversal à origem. Este sistema foi desenhado para oferecer uma experiência de utilização tão fácil e intuitiva quanto possível, permitindo também o armazenamento estável de grandes quantidades de informação e o tratamento das ferramentas incluídas de forma modular. Para tal, foi automatizado todo o processo de extração de informação de todas as ferramentas de análise, coordenado pelo motor de processamento. Após a extração, de acordo com o MDAV, é realizada a normalização, agregação e enriquecimento dos dados para sejam armazenados. O armazenamento local da informação, para além de permitir uma consulta muito mais rápida, permite não depender da disponibilidade de terceiros para consulta. Por fim, o motor de visualização, constituinte do sistema, possibilita a extração de relatórios que sumarizam a informação existente, agilizando eficazmente o processo de correção de vulnerabilidades.

A versão atual agrega informação extraída das três principais ferramentas utilizadas na gestão de vulnerabilidades interna. Estas são responsáveis pela análise dos ativos mais críticos da infraestrutura. Foram efetuados inquéritos às equipas técnicas da área da segurança informática, para avaliar a satisfação sobre os resultados produzidos. Para além dos inquéritos, foi ainda avaliada a capacidade de expansão do sistema, através da inclusão de uma nova ferramenta no CSVMS. Estes testes serviram para avaliar a robustez da solução, demonstrando elevado grau de satisfação com o resultado final.

6.2 Trabalho futuro

Relativamente ao trabalho futuro, acreditamos que os resultados obtidos são um ponto de partida benéfico ao melhoramento do processo de gestão de vulnerabilidades. O sistema atual é constituído por diversas tarefas e recursos que podem ser estudados e melhorados, tornando este processo ainda mais eficiente. Os próximos passos no sentido desta evolução incluem:

- **Desenvolvimento de um sistema de gestão de ativos críticos** - Este sistema seria responsável por catalogar os ativos da empresa de forma fácil e dinâmica. O sistema deve possibilitar a consulta de informação, podendo-se agregar dados complementares ao CSVMS. Desta forma, serviria como substituto ao repositório de enriquecimento atualmente utilizado.

- **Remoção de eventos duplicados** - Processar a informação recolhida das várias fontes de informação e agregar os dados que são reportados mais que uma vez para a mesma vulnerabilidade. O objetivo seria concentrar a informação das ferramentas numa só entrada do repositório, evitando a realização manual deste processo.

- **Integração de mais fontes de informação** - Integração de mais fontes de informação de forma a complementar ainda mais o repositório de conhecimento. Na vertente de ativos expostos à internet é de considerar a integração do *Shodan*[1] e *Probely*[2]. Já na análise de sistemas, podem ser integradas ferramentas como *Harpoon*[3] e *CIS-CAT.*[4]

[1] https://en.wikipedia.org/wiki/Shodan
[2] https://probely.com/
[3] https://sparkint.pt/
[4] https://www.cisecurity.org/cybersecurity-tools/cis-cat-pro/

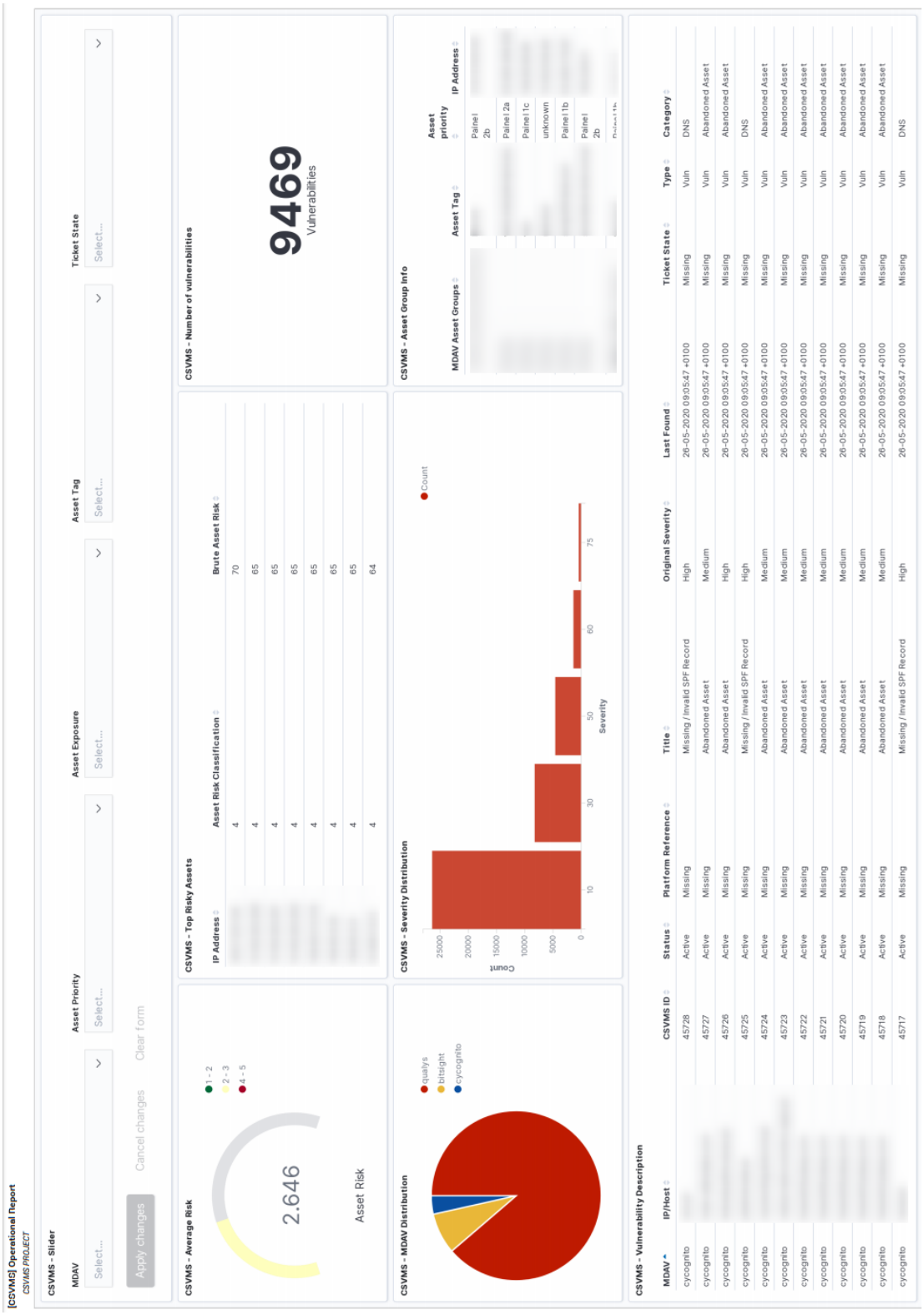
Apêndice A

Relatório operacional

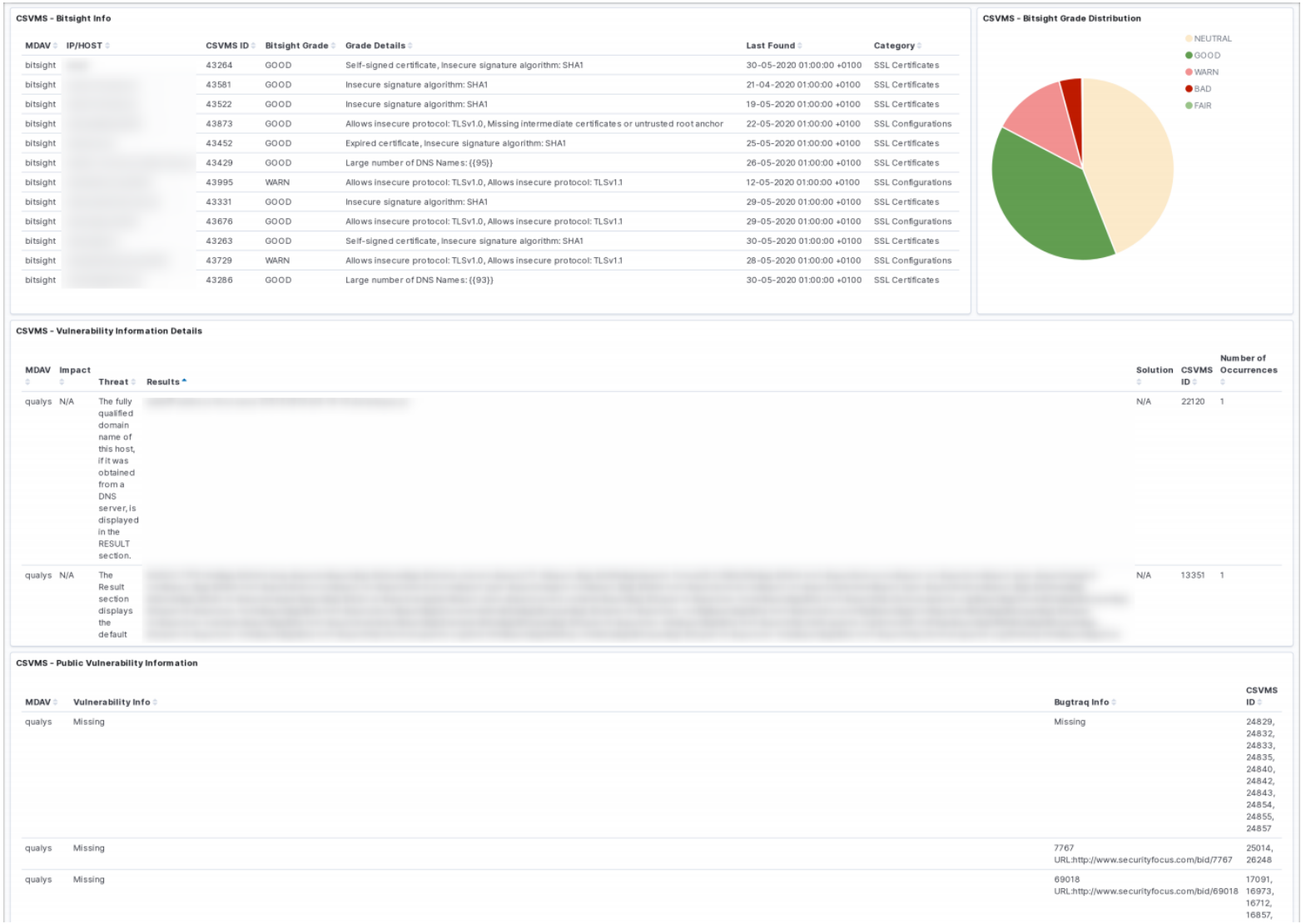
As duas figuras seguintes representam o relatório operacional produzido pelo CSVMS. Ainda que seja apenas um exemplo da disposição da informação do relatório original, a primeira parte permite consultar de forma mais generalizada o estado das vulnerabilidades da organização. Já na segunda parte, está presente a informação mais técnica relativa às vulnerabilidades encontradas.

Imagens nas próximas páginas

A.1 Exemplo de relatório operacional - parte 1



A.2 Exemplo de relatório operacional - parte 2



CSVMS - Bitsight Info

MDAV	IP/HOST	CSVMS ID	Bitsight Grade	Grade Details	Last Found	Category
bitsight	[redacted]	43264	GOOD	Self-signed certificate, Insecure signature algorithm: SHA1	30-05-2020 01:00:00 +0100	SSL Certificates
bitsight	[redacted]	43581	GOOD	Insecure signature algorithm: SHA1	21-04-2020 01:00:00 +0100	SSL Certificates
bitsight	[redacted]	43522	GOOD	Insecure signature algorithm: SHA1	19-05-2020 01:00:00 +0100	SSL Certificates
bitsight	[redacted]	43873	GOOD	Allows insecure protocol: TLSv1.0, Missing intermediate certificates or untrusted root anchor	22-05-2020 01:00:00 +0100	SSL Configurations
bitsight	[redacted]	43452	GOOD	Expired certificate, Insecure signature algorithm: SHA1	25-05-2020 01:00:00 +0100	SSL Certificates
bitsight	[redacted]	43429	GOOD	Large number of DNS Names: {{95}}	26-05-2020 01:00:00 +0100	SSL Certificates
bitsight	[redacted]	43995	WARN	Allows insecure protocol: TLSv1.0, Allows insecure protocol: TLSv1.1	12-05-2020 01:00:00 +0100	SSL Configurations
bitsight	[redacted]	43331	GOOD	Insecure signature algorithm: SHA1	29-05-2020 01:00:00 +0100	SSL Certificates
bitsight	[redacted]	43676	GOOD	Allows insecure protocol: TLSv1.0, Allows insecure protocol: TLSv1.1	29-05-2020 01:00:00 +0100	SSL Configurations
bitsight	[redacted]	43263	GOOD	Self-signed certificate, Insecure signature algorithm: SHA1	30-05-2020 01:00:00 +0100	SSL Certificates
bitsight	[redacted]	43729	WARN	Allows insecure protocol: TLSv1.0, Allows insecure protocol: TLSv1.1	28-05-2020 01:00:00 +0100	SSL Configurations
bitsight	[redacted]	43286	GOOD	Large number of DNS Names: {{93}}	30-05-2020 01:00:00 +0100	SSL Certificates

CSVMS - Bitsight Grade Distribution

CSVMS - Vulnerability Information Details

MDAV	Impact	Threat	Results	Solution	CSVMS ID	Number of Occurrences
qualys	N/A	The fully qualified domain name of this host, if it was obtained from a DNS server, is displayed in the RESULT section.	[redacted]	N/A	22120	1
qualys	N/A	The Result section displays the default	[redacted]	N/A	13351	1

CSVMS - Public Vulnerability Information

MDAV	Vulnerability Info	Bugtraq Info	CSVMS ID
qualys	Missing	Missing	24829, 24832, 24833, 24835, 24840, 24842, 24843, 24854, 24855, 24857
qualys	Missing	7767 URL:http://www.securityfocus.com/bid/7767	25014, 26248
qualys	Missing	69018 URL:http://www.securityfocus.com/bid/69018	17091, 16973, 16712, 16857,

Apêndice B

Questionário de usabilidade

B.1 Questionário de usabilidade adotado do modelo SUS

	Discordo totalmente				Concordo totalmente
1. Imagino-me a utilizar esta solução com frequência.					
	1	2	3	4	5
2. Achei a solução desnecessariamente complexa.					
	1	2	3	4	5
3. Achei a solução fácil de utilizar.					
	1	2	3	4	5
4. Preciso de apoio de um técnico para recorrer a esta solução.					
	1	2	3	4	5
5. Achei que as diversas funções da solução estavam bem integradas.					
	1	2	3	4	5
6. Achei que existia demasiada inconsistência na solução.					
	1	2	3	4	5
7. A maioria das pessoas irá aprender a utilizar a solução rapidamente.					
	1	2	3	4	5
8. Achei a solução demasiado pesada para ser utilizada.					
	1	2	3	4	5
9. Senti-me bastante confiante ao utilizar a solução.					
	1	2	3	4	5
10. Tenho de aprender muitos conceitos antes de poder utilizar a solução.					
	1	2	3	4	5

Abreviaturas

API Application Programming Interface.

CSOC Cyber Security Operations Center.

CSV Comma-separated Values.

CVE Common Vulnerabilities and Exposures.

CVSS Common Vulnerability Scoring System.

HTTP Hyper Text Transfer Protocol.

JSON JavaScript Object Notation.

MDAV Motor de Deteção e Análise de Vulnerabilidades.

NIAC National Infrastructure Advisory Council.

NIST National Institute of Standards and Technology.

OWASP Open Web Application Security Project.

RPA Robotic Process Automation.

SaaS Security as a Service.

SDI Sistema de Deteção de Intrusões.

SIEM Security Information Event Management.

SPAM Sending and Posting Advertisement in Mass.

SQLi SQL Injection.

TCP Transmission Control Protocol.

UDP User Datagram Protocol.

VAC Vulnerability Assessment Coordinator.

VACv2 Vulnerability Assessment Coordination Version 2.

XML Extensible Markup Language.

XSS Cross-site Scripting.

Bibliografia

[1] An overview of vulnerability scanners https://www.infosec.gov.hk/english/technical/files/vulnerability.pdf Acesso em 2020-03-15

[2] Common Vulnerabilities and exposures, CVE. http://cve.mitre.org/ Acesso em 2020-05-01

[3] Common Vulnerability Scoring System v3.0: Specification Document. //www.first.org/cvss/specification-document/ Acesso em 2020-05-01

[4] Nmap: The Network Mapper. https://nmap.org/ Acesso em 2020-01-07

[5] Fábio Guimarães Fernandes, Desenvolvimento de um processo automático de gestão de vulnerabilidades de ciber segurança em ambientes de grande dimensão, Tese de Mestrado em Engenharia Informática, FCUL, Setembro 2019.

[6] Vulnerability Scanning: What It Is and How to Do It Right. https://www.esecurityplanet.com/network-security/vulnerability-scanning.html Acesso em 2019-11-7

[7] What Is The CIA Triad? https://www.f5.com/labs/articles/education/what-is-the-cia-triad Acesso em 2019-10-7

[8] The Lifecycle of a Vulnerability. http://www.iss.net/documents/whitepapers/ISS_Vulnerability_Lifecycle_Whitepaper.pdf Acesso em 2019-10-26

[9] Elasticsearch - RESTful, Distributed Search and Analytics. https://www.elastic.co/pt/products/elasticsearch Acesso em 2020-06-01

[10] SANS - Implementing a Vulnerability Management Process. https://www.sans.org/reading-room/whitepapers/threats/implementing-vulnerability-management-process-34180 Acesso em 2019-11-2

[11] Astra - 4 Times Companies Were Forced to Shut Down Due to Hackers. https://www.getastra.com/blog/911/4-times-companies-were-forced-to-shut-down-due-to-hackers/ Acesso em 2019-10-23

[12] CYWARE - What is Cybersecurity Fingerprinting? https://securitytrails.com/blog/cybersecurity-fingerprinting Acesso em 2019-11-7

[13] Cloud Security Alliance -'The Treacherous Twelve' Cloud Computing Top Threats in 2016. https://cloudsecurityalliance.org/artifacts/the-treacherous-twelve-cloud-computing-top-threats-in-2016/ 2016 Acesso em 2020-01-19

[14] Wikipedia - Wannacry ransomware attack. https://en.wikipedia.org/wiki/WannaCry_ransomware_attack Acesso em 2019-11-08

[15] Wikipedia - Eternalblue. https://en.wikipedia.org/wiki/EternalBlue Acesso em 2019-11-08

[16] Wikipedia - DoublePulsar. https://en.wikipedia.org/wiki/DoublePulsar Acesso em 2019-11-08

[17] Impact of a vulnerability. https://www.intel.com/content/www/us/en/security-center/impact-of-vulnerability.htmlAcesso em 2019-11-8

[18] Gartner - Gartner Forecasts Worldwide Information Security Spending to Exceed $124 Billion in 2019.https://www.gartner.com/en/newsroom/press-releases/2018-08-15-gartner-forecasts-worldwide-information-security-spending-to-exceed-124-billion-in-2019 Acesso em 2019-11-25

[19] Why is Cyber Security Important.https://www.symantec.com/definitions/why-is-cyber-security-important Acesso em 2020-05-28

[20] IBM - X-force Exchange https://exchange.xforce.ibmcloud.com/ Acesso em 2019-12-5

[21] Pedro Miguel da Costa Santos, Continuous Security Assessment, Tese de Mestrado em Engenharia Informática, FCUL 2018.

[22] First - Common Vulnerability Scoring System v3.1: Specification Document, 2019 https://www.first.org/cvss/v3.1/specification-document Acesso em 2020-03-11

[23] OWASP - The Ten Most Critical Web Application Security Risks, 2019 https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf Acesso em 2020-01-09

[24] Elastic - Elasticsearch , 2019 https://www.elastic.co/pt/elasticsearch/ Acesso em 2020-06-12

[25] Elastic - Logstash : Collect, Parse, Transform Logs, 2020 https://www.elastic.co/pt/logstash Acesso em 2020-06-12

[26] Elastic - Kibana Explore, Visualize, Discover Data, 2020 https://www.elastic.co/pt/kibana Acesso em 2020-06-12

[27] BluePrism - Robotic Process Automation Software, 2019 https://www.blueprism.com/ Acesso em 2020-05-25

[28] ISO - iso 8061 Date and Time format https://www.iso.org/iso-8601-date-and-time-format.html Acesso em 2020-05-20

[29] SUS - A quick and dirty usability scale https://hell.meiert.org/core/pdf/sus.pdf Acesso em 2020-05-04